

**ANSI/NISO
Z39.50-1992
(Version 2)**

ANSI Z39.50: Information Retrieval Service and Protocol

**American National Standard
Information Retrieval
Application Service Definition
and Protocol Specification
for Open Systems Interconnection**

*Final Text
July 1992*

Abstract

This standard specifies an Open Systems Interconnection application layer service definition and protocol specification for Information Retrieval. The Information Retrieval protocol allows an application on one computer to query the database of another computer. The protocol specifies the procedures and structures for the intersystem submission of a search request (including the syntax of the query), request for the transmission of database records located by a search, the responses to the requests, access control, and resource control.

FOREWORD

This forward is not a formal part of the standard.

ANSI Z39.50-1992, *American National Standard Information Retrieval Application Service Definition and Protocol Specification for Open Systems Interconnection*, is a revision of ANSI Z39.50-1988. The 1988 and 1992 versions of Z39.50 are referred to as versions 1 and 2 respectively.

Version 1 was prepared by Committee D, "Computer-to-Computer Protocols", of NISO, the National Information Standards Organization. Committee D was organized to develop OSI application layer protocols for library applications, including an information retrieval protocol. The committee was disbanded after Z39.50 version 1 was approved. In 1989, the Z39.50 Maintenance Agency was formed, administered at the Library of Congress. Version 2 has been prepared by the Maintenance Agency, to incorporate enhancements described in detail below.

In 1990 the Z39.50 Implementors Group (ZIG) was established. Members include manufacturers, vendors, consultants, information providers, and universities, who wish to access or provide access to various types of information, including bibliographic, full text, financial, public utility, chemical, and news. Thus, although the protocol was originally proposed (in 1984) for use with bibliographic applications, interest in Z39.50 as an information retrieval protocol is no longer limited to use with bibliographic information.

Z39.50 version 2 represents a consensus of the ZIG, which has in effect acted in an advisory capacity to the maintenance agency, in the effort to develop version 2. ZIG Membership has been open to all interested parties.

Information Retrieval Protocol

The protocol specifies formats and procedures governing the exchange of messages between a requesting computer (the "origin") and a responding computer (the "target") to enable the origin to 1) request that the target search a database and identify records which meet specified criteria, and 2) request transmission of, and receive, some or all of the identified records.

The origin may initiate requests on behalf of a user who wishes to search a database located on the target system. The protocol addresses communication between corresponding information retrieval

applications on the origin and target systems; it does not address interaction between the origin computer and user.

Basic Operations

The Information Retrieval protocol provides the following basic capabilities. The origin may submit a Search request which includes a query, and parameters which determine whether or not records resulting from the search are to be returned as part of the response. The target responds with a count of records identified and possibly some or all of the records. The origin may then submit a Present request, requesting transmission of selected records. The origin assumes that records selected by the search request form an ordered set (the "result set") which may be referenced by sequential position within the set. Record order is determined by the target. The origin may request (for example) records one through four, and follow with a request for records four through eight, and then records two through three, etc. The origin may submit as many such Present requests as desired, and may then submit another Search request.

Optional capabilities include the following:

- The origin may specify element set names indicating data elements which should be transmitted in cases where the origin does not wish to receive complete database records. For example, the origin might specify "If 5 or less records are identified, transmit 'full' records; if more than 5 records are found, transmit 'brief' records". (In this example, the meanings of 'full' and 'brief' are target specific.)
- The origin may specify a preferred-record-syntax for response records. For example, it might identify a bibliographic format such as USMARC.
- The origin may name a result set for subsequent reference.
- The origin may delete a named result set.
- The target may impose access control restrictions on the origin, by demanding authentication before processing a request.
- The target may provide resource control by sending an unsolicited or solicited status report. This might occur during processing of an operation, in which case the target may

suspend processing and allow the origin to decide whether the operation should continue.

Query Formulation

This standard mandates support of the "Type-1" query as a common format for the intersystem transmission of a search query. The "Type-1" query, fully specified in this standard, expresses a search query by individual search terms (or phrases) with a set of attributes for each. Attributes may specify, for example, type of term (subject, name, etc.), whether it is truncated, and its structure. The target is responsible for mapping these attributes to the logical design of the database.

Several terms may be combined in a Type-1 query, linked by boolean operators. Terms and operators are expressed in Reverse Polish Notation.

Attribute Sets

The attributes associated with a search term belong to a particular attribute set, whose definition assigns integer values to various attributes. The definition is assigned a unique and globally recognized "attribute-set-id", an OSI Object Identifier, which is included within the query.

The process of assigning an OSI object identifier is called "registration". Appendix C (which is part of the standard) defines and registers one attribute-set-id, "bib-1", which specifies various attributes that are useful for bibliographic queries. There is a mechanism to register additional attribute sets. Just as the bib-1 attribute set was developed by the bibliographic community, it is intended that attribute sets will be developed and registered as needed by other communities (for example, a chemical attribute set by chemists).

Response Records

The protocol provides for the transfer of database records identified by a Search or Present request. The standard distinguishes two types of records, database and diagnostic records, which can occur in response messages from the target.

Appendix E (which is part of this standard) registers object identifiers for various MARC formats, including USMARC, UKMARC, Norway MARC and CANMARC; these object identifiers accompany database record returned by the target. There is a provision for registration of additional record formats, including non-bibliographic formats.

Diagnostic records are similarly accompanied by an object identifier which identifies their format. Appendix D (which is part of this standard) defines

and registers one diagnostic record format, "bib-1", which includes various diagnostic codes that are useful for bibliographic applications. Additional diagnostic record formats may be registered.

Changes in Version 2

Z39.50 version 2 incorporates major enhancements to version 1, in two categories:

- changes necessary for alignment with ISO 10162/10163, the Search and Retrieve (SR) Service Definition and Protocol Specification; and
- incorporation of features deemed necessary by implementors, to provide sufficient functionality so that implementation can be economically justified.

Alignment with SR

When SR was approved in 1991, incompatibilities between Z39.50 version 1 and SR remained:

- SR Search and Present requests include parameters (related to record composition and syntax) not included in Z39.50 version 1. Specifically: (1) the search request in SR has both small-set- and medium-set- element-set-name parameters. Z39.50 version 1 has only a single, or global, element-set-name parameter. (2) SR includes the parameter Preferred-record-syntax, and Z39.50 version 1 does not.
- The SR Delete service allows the origin to specify a list of result sets to be deleted. Z39.50 version 1 allows deletion of a single specified result set only.
- SR uses ASN.1 ("Abstract Syntax Notation one") to define the abstract syntax of its protocol data units, and specifies ASN.1 Basic Encoding Rules for transfer syntax. Z39.50 Version 1, finalized before ASN.1 was stable, developed a non-standard abstract syntax notation and corresponding transfer syntax, which in version 2 have been abandoned in favor of ASN.1.
- A major limitation of Z39.50 version 1 is the lack of flexibility to reference objects, including attribute-sets, diagnostic-sets, and record formats, which SR references through OSI object identifiers.

Z39.50-Specific Features

The above differences are resolved in Z39.50 version 2. There will still be differences with SR, however. Z39.50 includes two services not yet in SR: access control and resource control. Some U.S.

information providers cannot use Z39.50 if it does not include an access control service. Both services have been carefully incorporated so that an SR and Z39.50 version 2 origin/target pair may interwork transparently (i.e. the SR implementation would not be aware that its partner is a Z39.50 rather than SR implementation).

In general, implementors who plan to build Z39.50-specific features into their implementations, for use when interworking with a Z39.50 partner, would suppress these features when interworking with an SR partner. Thus, lack of these features in SR will not inhibit interoperability, although it might limit potential functionality. For example, information that an SR origin can receive from a Z39.50 target might be limited because the origin cannot respond to an access control request. (The same would be true of a Z39.50 origin which does not support access control -- it is an optional service). The limitation would not owe to an incompatibility between origin and target implementations, but rather, because the target requires authentication and the origin cannot provide it.

Version 2 includes changes to the Resource Control facility. The origin will be able to request a resource control action or cancel an operation. These changes have been expressed as requirements by the ZIG.

Version 2 specifies five query types. Types 0, 1, and 2 are identical to the three query types specified in SR. The "Type-0" query is designated "private", allowing two systems to use a private, mutually agreed upon query format. "Type-1" is described above. "Type-2" is as specified by ISO 8777. Z39.50 specifies two additional query types: "Type-100" is as specified by ANSI Z39.58, and "type-101" is for proximity searching. The latter is specified in appendix G. Proximity searching is a feature that implementors recommended be incorporated into this version.

Object Identifiers

Z39.50 registers all information objects that are registered in SR: application context, abstract syntax, attribute set, diagnostic set, and record syntax definitions. In addition, Z39.50 attribute set and diagnostic set definitions contains additional attribute and diagnostic values which are "U.S. specific".

Z39.50 registers two additional object identifier classes: resource report format (for resource control, which is not yet in SR), and transfer syntax. The latter accommodates non-bibliographic databases. Z39.50 also provides a mechanism for registration of

experimental and implementation-specific objects. The Z39.50 Maintenance Agency will act as registration authority for Z39.50 objects.

Future Services

The generality of the protocol is intended to allow accommodation of new services as they are required. Some services being considered are:

- **Explain:** to allow an origin to obtain details of the target implementation, including databases available for searching, data elements in records of a specified database, element sets supported, elements within in a specified element set, attribute sets supported, attributes supported within a supported attribute set, diagnostic sets, and record syntaxes.
- **Define Element Set:** to allow an origin to define and name a set of elements to compose a retrieval record.
- **Scan:** to allow an origin to obtain a list of access point values from a database index (subject terms, names, titles, etc.), preceding and following a specified access value.
- **Object Access:** To allow an origin to perform operations including create, delete, sort, export, activate, and permit, on various objects, including result sets and queries.
- **segmentation:** to allow a target to return multiple responses to a single present request.

TABLE OF CONTENTS

1. Introduction	1
1.1 Scope and Field of Application	1
1.2 Model	1
1.3 References	2
2. Definitions	2
3. Information Retrieval Service	3
3.1 General characteristics of the Information Retrieval Service	3
3.2 Facilities of the Information Retrieval Service	4
3.2.1 Initialization Facility	4
3.2.2 Search Facility	6
3.2.3 Retrieval Facility	11
3.2.4 Result-set-delete Facility	12
3.2.5 Access Control Facility	13
3.2.6 Accounting/Resource Control Facility	14
3.2.7 Termination Facility	17
3.3 Message Size Limitations	17
3.4 Reference-id	18
4. Protocol Specification	18
4.1 Abstract Syntax of the Information Retrieval Protocol	19
4.2 Protocol Procedures	27
4.2.1 Services Required	27
4.2.2 Protocol Model	27
4.2.3 State Tables	28
4.2.4 Protocol Errors	29
4.3 Rules for Extensibility	29
4.4 Conformance	33
APPENDIX A: Object Identifiers Assigned in This Standard	34
APPENDIX B: Definition of Application Context basic-Z39.50-ac	34
APPENDIX C: Attribute Set bib-1	35
APPENDIX D: Diagnostic Set bib-1	37
APPENDIX E: Record Syntaxes	39
APPENDIX F: Resource Report Format bib-1	40
APPENDIX G: Proximity Query	41

1. Introduction

This is one of a set of standards produced to facilitate the interconnection of computer systems. It is positioned with respect to other related standards by the Open Systems Interconnection (OSI) basic reference model (ISO 7498). The aim of Open Systems Interconnection is to allow the interconnection of computer systems, with a minimum of technical agreement outside the interconnection standards.

This standard defines a protocol within the application layer of the reference model, and is concerned in particular with the retrieval of information stored in machine readable databases.

1.1 Scope and Field of Application

This standard describes the Information Retrieval application service (section 3) and specifies the Information Retrieval application protocol (section 4), for Open Systems Interconnection.

The Information Retrieval application service is described in terms of services which provide capabilities within an application. The description neither specifies nor constrains the implementation within a computer system. The purpose of the service description is to define the functions that the protocol must support.

The protocol specification includes the definition of the protocol control information, the rules for exchanging this information, and the conformance requirements to be met by implementation of this protocol.

This standard is intended particularly for use by systems supporting information retrieval services for organization such as libraries, information utilities, and union catalogue centers. It addresses connection-oriented, program-to-program communication utilizing telecommunications. It does not address the interchange of information with terminals or via other physical media.

1.2 Model

The objective of this standard is to facilitate the open interconnection of database users with database providers. It is necessary to distinguish between the set of OSI standards and the hardware and software implementation of a system using the protocols specified in these standards. The ways in which databases are implemented differ considerably; different systems have different styles for describing the storage of data and the means by which it can be accessed. A common, abstract model is therefore used in describing databases, to which an individual

system can map its implementation. This enables different systems to communicate in standard, mutually understandable terms.

The term "database" as used in this standard refers to a collection of one or more files, each with a unique name. A group of files within a database may also have a name and be accessed as a single database. The unit of information for retrieval from a database is a record. All of the records within a given file have a common structure, contain a common set of data elements and a common set of access points. An access point is a unique or non-unique key which can be specified either singly or in combination with other access points in a search for records. An access point may be equivalent to a data element, be derived from a data element, or the combination of all or part of two or more data elements.

A search query may be applied to a database, specifying values to be matched against the access points of the database. The subset of records formed by applying the search query is termed the result set. A result set may itself be referenced in a subsequent search query statement and manipulated to form a new result set.

For generality, it is assumed that query processing does not necessarily require physical access to records; a result set is thus assumed to be the identification of (e.g. pointers to) records, as opposed to the actual set of records, selected by a query. (It is not assumed that the database records are locked. Methods of concurrency control, which would prevent modification or deletion of result set records, are not addressed by this standard.) A result set may be used as a selection mechanism for the transfer of records between systems; the result set itself is considered to be a purely local data structure and is not transferred (that is, records are transferred, but not the local pointers to the records).

A generic search query statement is composed of a database name followed by a query statement. The Type-1 query statement defined in this standard consists of either a single access point clause, or several access point clauses linked by logical operators. For example:

"In the database which is named 'Books' find all of the records for which the access point 'title word' contains the value 'evangeline' AND the access point 'author' contains the value 'long-fellow'".

Following the processing of a search, the set of items with the specified properties (the result set) is made available by the target system, to the origin system, for subsequent retrieval requests. For the

purpose of retrieving records, the logical structure of a result set is that of a named, ordered list of triples consisting of (a) an ordinal number corresponding to the position of the triple in the list, (b) a database name, and (c) a unique record identifier (of local significance only) within the database named in (b). A result set item is referenced by its position within the result set, that is, by (a).

1.3 References

- ANSI Z39.58 -- *Common Command Language for Online Interactive Information Retrieval* 1991
- ISO 2709 -- *Documentation - Format for Bibliographic Information Interchange on Magnetic Tape* 1981.
- ISO 4217 - *Codes for the representation of currencies and funds* 1990.
- ISO 7498 -- *Information Processing Systems - Open Systems Interconnection - Basic Reference Model* 1984.
- ISO TR 8509 -- *Information Processing Systems - Open Systems Interconnection - Service Conventions* 1987.
- ISO 8649 -- *Information Processing Systems - Open Systems Interconnection - Service Definition for the Association Control Service Element* 1987.
- ISO 8650 -- *Information processing systems - Open Systems Interconnection - Protocol Specification for the Association Control Service Element* 1987.
- ISO 8777 -- *Documentation - Commands for Interactive Text Searching*
- ISO 8822 -- *Information Processing Systems - Open Systems Interconnection - Connection Oriented Presentation Service Definition* 1988.
- ISO 8824 -- *Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)* 1987.
- ISO 8825 -- *Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)* 1987.
- ISO 9545 -- *Information processing systems - Open Systems Interconnection - Application Layer Structure* 1989.
- ISO 10163 -- *Documentation - Search and Retrieve Protocol Specification* 1991.

2. Definitions

For terms which are formally defined in other standards, the formal definition is given, italicized, and the standard is listed in parentheses. In those

cases, descriptions and/or alternate definitions (indented) are sometimes provided. All of the definitions below including the alternate definitions apply only to the Information Retrieval application service and protocol, and only within the context of this standard.

APDU - See Application Protocol Data Unit

Application Association -- *A cooperative relationship between two application-entity-invocations for the purpose of communication of information and coordination of their joint operation. This relationship is formed by the exchange of application-protocol-control-information using the Presentation-Service.* (ISO 9545)

For the Information Retrieval service and protocol, an application association is analogous to an individual communication session between a database user and a database provider. Each association consists of an origin application and a target application, and these roles may not be reversed within an association.

Application Entity -- *The aspects of an application-process pertinent to OSI.* (ISO 7498)

Application Protocol -- The rules governing the format and exchange of information between an origin and target application.

Application Protocol Control Information -- The information conveyed by application protocol data units.

Application Protocol Data Unit -- *A unit of data specified in an application-protocol and consisting of application-protocol-information and possibly application-user-data.* (ISO 7498)

A unit of data passed between an origin and a target.

Application Service User -- That portion of an application which makes requests upon the open systems environment. (The concept of service-user is employed to facilitate the specification of protocol procedures and is not analogous to the database user.)

ASN.1 -- Abstract Syntax Notation One.

Conditionally Confirmed Service -- A confirmed service in which the response might under certain conditions not occur.

Confirmed Service — *A distinct part of the total IR-service which results in an explicit confirmation from the service-provider to the initiating service-user.* (ISO TR 8509)

A service which consists of a request followed by a response. A confirmed service may be initiated by the origin, in which case it consists of a request by the origin followed by a response from the target; or it may be initiated by the target, in which case it consists of a request by the target followed by a response from the origin.

Connection Oriented Communication — Communication in which the communication path is explicitly established for an association, maintained throughout the association, and explicitly terminated.

Database Provider — The application which provides access to a database local to that application.

Database User — The application which accesses a remote database.

IR — Information Retrieval

Origin Application — The application that initiates an association and is the source of requests during the association. The database user.

Name — A linguistic construct, expressed in some language, which corresponds to an object. A name denotes (i.e. identifies) the object to which it is bound.

Non Confirmed Service — *A distinct part of the total IR-service which does not result in an explicit confirmation from the service-provider to the initiating service-user.* (ISO TR 8509)

A service which consists of a request only (not followed by a response). It may be initiated by the origin or the target.

OSI — Open Systems Interconnection.

Primitive — *An abstract, implementation-independent representation of an interaction between the service-user and the service-provider.* (ISO TR 8509)

Result set — An ordered list of triples consisting of (a) an ordinal number corresponding to the position of the triple in the list, (b) a database name, and (c) a unique record identifier (of local significance only)

within the database named in (b). A result set is formed by applying a search query.

RPN — Reverse Polish Notation

Service provider — *An abstract of the totality of those entities which provide a service to peer service-users.* (ISO TR 8509)

(Note: the concept of service-provider is employed to facilitate the specification of protocol procedures. It is not analogous to the database provider, and it does not refer to providers of telecommunication services.)

Target Application — The application that accepts an association and is the sink for requests during the association. The database provider.

Primitive Name — A kind of name, the internal structure of which is not required to be understood or have significance to users of the name.

3. Information Retrieval Service

This section defines the Information Retrieval service, which is supported by the Information Retrieval protocol.

3.1 General characteristics of the Information Retrieval Service

The service definition describes an activity between two applications on different computers: an initiating application, the origin, and a responding application, the target. The target is associated with one or more databases. Communication between origin and target is via an application association. An association is explicitly established by the origin and may be explicitly terminated by either origin or target, or implicitly terminated by a communication failure or other external event.

The roles of origin and target may not be reversed within an association. An association cannot be restarted, thus no status information is retained once an association is released.

The complete application service is composed of the association control service element, which provides association management, and one or more specific application services, such as the Information Retrieval application service. There are three distinct phases during the life of an application association: establishment, information transfer, and termination. The association control service element provides all of the services required during the establishment and

termination phases, including the selection of an application context specifying, among other things, the set of service elements which are valid during the information transfer phase. Section 4.2.1.2 specifies those services for association control which are assumed by the Information Retrieval service.

3.2 Facilities of the Information Retrieval Service

There are seven facilities defined by this standard. Each of the Initialization, Search, Present, Result-set-delete, and Access Control facilities consists of a single service. The Accounting/ Resource Control facility consists of three services. The Termination facility consists of two services.

- (1) **Initialization Facility Init Service:** Init request by the origin followed by an Init response from the target. The Init service is a confirmed service initiated by the origin.
- (2) **Search Facility Search Service:** Search request by the origin followed by a Search response from the target. The Search service is a confirmed service initiated by the origin.
- (3) **Retrieval Facility Present Service:** Present request by origin followed by a Present response from the target. The Present service is a confirmed service initiated by the origin.
- (4) **Result-set-delete Facility Delete Service:** Delete request by the origin followed by a Delete response from the target. The Delete service is a confirmed service initiated by the origin.

Note: Between the Init, Search, Present, or Delete request and its respective response, there may occur one or more of the following: Access-control request/-response sequence initiated by the target, Trigger-resource-control request by the origin, and Resource-control request (and possibly response) initiated by the target.

- (5) **Access Control Facility Access-control service:** Access-control request by the target, following an Init, Search, Present, Delete, or Resource-report request by the origin and followed by an Access-control response from the origin. The Access-Control service is a confirmed service initiated by the target.
- (6) **Accounting/Resource Control Facility** The Accounting/ Resource Control facility consists of three services: the Resource-control service, the Trigger-resource-control service, and the Resource-report service.

- **Resource-control Service:** Resource-control request by the target, following an Init, Search, Present, or Delete request by the origin and followed (possibly) by a Resource-control response from the origin. The Resource-control service is a conditionally confirmed service, initiated by the target.
- **Trigger-resource-control Service:** Trigger-Resource-control request by the origin, following an Init, Search, Present, or Delete request by the origin. The Trigger-resource-control Service is a non-confirmed service initiated by the origin.
- **Resource-report Service:** Resource-report request by the origin, following an Init, Search, Present, Delete or Resource-report response from the target. The Resource-report service is a confirmed service, initiated by the origin.

Note: Between the Resource-report request and its response, there may occur one or more Access-control request/response sequences.

- (7) **Termination Facility** The Termination Facility allows an origin or target system to initiate abrupt termination of the association, or an origin system to initiate graceful termination.
 - **IR-abort Service:** IR-abort request by either the origin or the target. The IR-abort Service is a non-confirmed service initiated by either the origin or target.
 - **IR-Release Service:** IR-release request by the origin followed by an IR-release response by the target. The IR-Release Service is a confirmed service initiated by the origin.

The IR-abort and IR-release services map directly onto the A-ABORT and A-RELEASE services (respectively) of the association control service element.

3.2.1 Initialization Facility

3.2.1.1 Init Service

The Init service allows an origin to propose values for initialization parameters. The target system may propose alternative values for some of the parameters. If so, the origin must either accept the alternative values proposed by the target or else terminate communication.

3.2.1.1.1 Id/authentication The origin and target agree, outside the scope of the standard, whether or not this parameter is to be supplied by the origin, and

<u>Parameter</u>	<u>Origin Request</u>	<u>Target Response</u>
Id/authentication	x (optional)	
Options	x	x
Preferred-message-size	x	x
Maximum-record-size	x	x
Result		x
User-information-field	x (optional)	x (optional)
Reference-id	x (optional)	x (if applicable)

Table 1: *Parameters of the Init Service*

if so, what the value is. This value is used by the target to determine if the origin is authorized to enter into communication with the target.

3.2.1.1.2 Options The Init request specifies either "will use" or "will not use", and the Init response specifies "will support" or "will not support" for the following capabilities:

1. search
2. present
3. delete
4. resource-report
5. trigger-resource-control

If the target specifies "will support" for a particular capability, then the origin may assume that the service will be available and may use the service during the association, even if the origin had specified "will not use" for that service.

In addition, the Init request specifies either "will support" or "will not support", and the Init response specifies "will use" or "will not use" for each of the following capabilities:

1. resource-control
2. access-control

If the request specifies "will not support" for a given capability, and the response specifies "will use" for that capability, then the value of Result must be "reject".

Note: the above two lists of capabilities are subject to expansion in future versions of this protocol.

search - The origin specifies "will use" for "search" if it intends to submit Search requests.

The target indicates that it is willing (or unwilling) to accept and respond to Search requests by specifying "will support" (or "will not support") for "search".

present - The origin specifies "will use" for "present" if it intends to submit Present requests. The target indicates that it is willing (or unwilling) to accept and respond to Present requests by specifying "will support" (or "will not support") for "present".

delete - The origin specifies "will use" for "delete" if it intends to submit Delete requests. The target indicates that it is willing (or unwilling) to accept and respond to Delete requests by specifying "will support" (or "will not support") for "delete".

Resource-report - The origin specifies "will use" for "resource-report" if it intends to submit Resource-report requests. The target indicates that it is willing (or unwilling) to accept and respond to Resource-report requests by specifying "will support" (or "will not support") for "Resource-report".

Note: The target indication that it is willing to respond to Resource-report requests does not imply that it will include a resource report in the response.

Trigger-resource-control - The origin specifies "will use" for "trigger-resource-control" if it intends to submit Trigger-resource-control requests (if so, the origin must also indicate that it is prepared to receive and respond to a Resource-control request from the target, by specifying "will support" for "resource-control"). The target indicates that it is willing (or unwilling) to accept Trigger-resource-control requests by specifying "will support" (or "will not support") for "trigger-resource-control".

Notes:

- (1) The target may indicate unwillingness to accept Trigger-resource-control requests even if it specifies "will use" for "resource-control".
- (2) An indication by the target of willingness to accept Trigger-resource-control requests does not imply that the target will take any action as a result of a Trigger-resource-control request.

resource-control - The origin indicates that it is prepared to receive and respond to a Resource-control request from the target, by specifying "will support" for "resource-control". Conversely, the origin indicates that it is not prepared to receive a

Resource-control request by specifying "will not support". In the latter case, if the target cannot suppress sending a Resource-control request, it should reject the connection by setting Result to "reject", specifying "will use" for "resource-control", and (optionally) supplying a text message in the User-information-field.

access-control - Likewise, the origin indicates whether or not it is prepared to receive and respond to an Access-control request from the target, by specifying "will support" or "will not support" for "access-control".

Security is invoked at different levels. In addition to user authentication at the outset of an association, security might be invoked to control access to a particular database, record, result-set, or use of a command. It might be used to determine whether an origin is authorized to request a resource report.

If the origin is not capable of receiving an Access-control request, and if security requirements on the target system mandate that security (other than that which might be provided by the parameter Id/authentication) be invoked at the outset of an association, then the target should reject the association by setting the parameter Result to "reject", and specifying "will use" for "access-control". However, if the target invokes security, but not at the association level, then the target may choose to accept the connection. In that case, if the target subsequently receives a command that would precipitate an Access-control request, the target agrees to suppress the request and respond to the command with an error status indicating that a security challenge was required but could not be issued.

3.2.1.1.3 Preferred-message-size and Maximum-record-size The Init request contains Preferred-message-size and Maximum-record-size, specified in bytes. Maximum-record-size must be greater than or equal to preferred-message-size. The Init response contains both the Preferred-message-size and Maximum-Record-size that the target is going to use.

The target has the option of responding with values different from those requested by the origin (however, Preferred-message-size must be less than or equal to Maximum-record-size), in which case, the origin may abort the connection if those values are not acceptable. The usage of these parameters is specified in section 3.3.

3.2.1.1.4 Result The target indicates whether or not it accepts the association by specifying a value of "accept" or "reject" respectively in the parameter Result. If "reject" is indicated, the origin is expected to terminate communication.

3.2.1.1.5 User-information-field This field may be used by either the origin or target for additional information, not specified by this standard.

3.2.1.1.6 Reference-id See section 3.4.

3.2.2 Search Facility

The Search facility enables an origin system to query databases at a target system, and to receive information about the results of the query.

3.2.2.1 Search Service

The Search service allows an origin to send a query to a target. The basic query concept is: "from the specified set of items, identify those with the properties indicated." The set of items identified is called the "result set", and is maintained by the target for subsequent retrieval requests. However, depending on the parameters of the search, one or more items identified by the result set may be immediately returned as part of the search response. The result set is an ordered set; items identified by entries in the result set are referenced by the position of the entry within the result set, beginning with one (1).

3.2.2.1.1 Query-type and Query The parameter Query-type identifies the syntax of the query. As noted above, the basic query concept is "from the specified set of items, identify those with the properties indicated." The "specified set of items" is a collection of one or more databases, specified by the parameter database-names. The "properties indicated" are specified by the parameter Query.

Five types are defined:

- type-0 may be used only when the origin and target have an a priori agreement outside of the standard;
- type-1 is the Reverse Polish Notation (RPN) query specified in section 3.2.2.1.1.1;
- type-2 is the ISO8777 type query, specified in ISO 8777;
- type-100 is the Z39.58 type query, specified in ANSI Z39.58; and
- type-101 is the proximity query, specified in appendix G.

<u>Parameter</u>	<u>Origin Request</u>	<u>Target Response</u>
Query-type	x	
Query	x	
Database-names	x	
Result-set-name	x	
Replace-indicator	x	
Small-set-element- set-names	x (opt.)	
Medium-set-element- set-names	x (opt.)	
Preferred-record-syntax	x (opt.)	
Small-set-upper-bound	x	
Large-set-lower-bound	x	
Medium-set- present-number	x	
Database/ diagnostic-records		x (if appl.)
Result-count		x
Number-of-rec- ords-returned		x
Next-result-set-position		x
Search-status		x
Result-set-status		x (if appl.)
Present-status		x (if appl.)
Reference-id	x (opt.)	x (if appl.)

Table 2: Parameters of the Search Service

3.2.2.1.1.1 RPN Query This section specifies procedures when Query-type is 1, 'RPN-query'. The RPN query has the following structure:

```

RPN-query ::= argument |
            argument + argument + operator
argument  ::= operand | RPN-query
operand   ::= attribute-set + term |
            Result-set-id
operator  ::= AND | OR | AND-NOT
Where    ::= means "is defined as",
            | means "or",

```

+ means "followed by", and + has precedence over | (i.e., + is evaluated before |).

Notes:

- (1) "RPN-query" is recursively defined; it is either
 - a) "operand" , or
 - b) "argument + argument + operator".
 In case (b), each occurrence of "argument" can be replaced by either (a) or (b) and so on. A structure composed of operators and operands conforms to the Type-1 query syntax if and only if it is possible, by repeatedly replacing occurrences of (b) with (a), to reduce the structure to (a).
- (2) "operand" is either (a) attribute-set + term, or (b) Result-set-id. In either case it represents a set of database records. For (a) it is the set of database records obtained by evaluating the specified attribute-set and term against the collection of databases specified in the Search request (see Note 3). For (b) it is the set of database records represented by the result set for which Result-set-id was specified as the value of the parameter Result-set-name in a previous Search request.
- (3) Different attribute sets will be defined and registered (Appendix C defines and registers attribute-set bib-1). An example of an attribute-set/term combination is 'title word'/'evangeline'; in this case, evaluation of attribute-set/term against a database would result in the identification of all of the records in the database for which the access point 'title word' contains the value 'evangeline'.

Representation and Evaluation of the Type-1 Query

At the origin system, the Type-1 query is represented by a tree. Each leaf node contains a simple operand and each non-leaf node contains a complex operand. A simple operand is either a Result-set-id or an Attribute-set+Term. A complex operand is a subtree whose root is an operator and which contains two subtrees: a left-hand operand and a right-hand operand, LO and RO. A complex operand is thus one of the following:

- LO AND RO, representing the intersection of LO and RO;
- LO OR RO, representing the union of LO and RO; or
- LO AND-NOT RO, representing the set of elements in LO that are not in RO.

At the target, evaluation of the tree is illustrated by the use of a stack. The tree is processed according to a left post-order traversal. Each leaf-node is one of the following:

- a) Attribute-set + Term
- b) Result-set-id
- c) Operator

Whenever (a) is encountered, it is evaluated against the collection of databases specified in the Search request, and the result is put on the stack. Whenever (b) is encountered, it is put on the stack.

Whenever (c) is encountered, the last two items (i.e. sets, see Note 2 above) that have been put on the stack are pulled off and the operator is applied as follows:

- if operator is AND, the result is the intersection of the two sets,
- if operator is OR, the result is the union of the two sets,
- if operator is AND-NOT, the result is the set of elements in the first set (i.e., the first of the two sets to have been placed on the stack) which are not in the second set.

The resulting set is then put on the stack.

When evaluation of the query is complete (i.e. all query-terms have been processed) there will be one item remaining on the stack (otherwise the query is in error) which is the result of the query.

The following examples illustrate query evaluation. In these examples, D represents the collection of databases specified in the Search request, R represents a Result-set-id, and A, B, and C represent attribute-list/term combinations such as "subject = dogs".

1. Query = A
Result: the records in D for which A is true
2. Query = A B C AND OR
Result: the records in D for which both B and C are true, or A is true
3. Query = A B AND C OR
Result: the records in D for which both A and B are true, or C is true
4. Query = R A AND
Result: the records in D for which both (1) A is true, and (2) which are also in result set R
5. Query = R A OR
Result: the records in D for which A is true, together with records in R

3.2.2.1.2 Database-names The target designates, by agreement outside the scope of the standard, what databases may be specified on a Search request, and also in what combinations they may be specified.

For example, a target might specify that databases A, B, and C, may be searched individually, and that A and B may be searched in combination (but not A and C, nor B and C). If an origin requests a combination of databases which is not supported, the search will result in a diagnostic such as "combination of specified databases not supported" (see appendix D).

3.2.2.1.3 Result-set-name and Replace-indicator The parameter Result-set-name specifies a name to be given to the result set which will be created by the query so that it may be subsequently referenced within the same association.

- o If a result set with the same name already exists at the target, the action taken depends on the value of the parameter Replace-indicator, as follows:
 - If the value of Replace-indicator is "on", then after processing the query, the existing result set whose name is specified by the parameter Result-set-name will be deleted, and a new result set by that name created. If the search cannot be processed, the content of the result set will be empty.
 - If the value of Replace-indicator is "off", the search is not processed, an error diagnostic is returned by the target, and the existing result set whose name is specified by the parameter Result-set-name is left unchanged.
- o If a result set does not exist with the name specified by the parameter Result-set-name, then a result set by that name is created by the target, and the parameter Replace-indicator is ignored. The initial content of the result set is empty. If no records are found by the query, the result set remains empty.

A target system need not support, in general, the naming of result sets by the origin (see however section 4.4.3, "Statement Requirements" for Conformance). However, a target system must support at least the result set whose name is "default". If the origin specifies "default" as Result-set-name, then Replace-indicator must be "on".

A result set which is created by a Search request (that is, specified by the parameter Result-set-name) may be referenced in a subsequent Present request or as an operand in a subsequent Search request (for example, in a Type-1 query). If a result set named "default" is created, it remains available for reference from the time it is created until the end of the association during which it is created, or until either:

- another "default" result set is created, because the name "default" is specified as Result-set-name in a subsequent Search request, or
- it is unilaterally erased or deleted by the target.

Any result set other than the result set named "default" remains available for reference from the time it is created until it is deleted in one of the following ways:

- by a Delete request;
- implicitly, because a result set was specified by the same name in a Search request, and the value of the parameter Replace-indicator was "on";
- unilaterally by the target (at any time); or
- by termination of the association.

3.2.2.1.4 Small-set-element-set-names and Medium-set-element-set-names An element set name is a primitive name which specifies a particular subset of the elements in a database record which are to compose the response records. The specified subset is made up of components of the abstract-syntax description of the database records and is, therefore, a formal subset of that abstract-syntax-definition. Element set names are specified, along with their definitions, for a given database, by the target, outside of this standard. The target specifies a default element set for each database.

The parameters Small-set-element-set-names and Medium-set-element-set-names describe the preferred composition of the records expected in the search response. If the query results in a small-set (see 3.2.2.1.6), then Small-set-element-set-names pertains. If the query results in a medium-set, then Medium-set-element-set-names pertains.

Each of the two parameters is a set of one or more pairs of a database name and associated element set name. For each database record returned in a Search (or Present) response, if the given database is specified (as a component of one of the pairs comprising the pertaining element set name) then the response record should be composed according to the corresponding element set name. If not, the response record should be composed according to the default element set name for that database. If an element set name is specified which is not valid for the corresponding database, then the Search Response will include a diagnostic, such as "element set name not valid for database", and the value of the parameter Search-Status will be "failure".

Each of the two parameters may alternatively consist of a single element set name (from those

defined by the target system), with no database specified. In that case, for each database record returned in a Search (or Present) response:

- if the specified element set name is valid for the given database, the response-record should be composed according to that element set name;
- if the specified element set name is not valid for the given database, the response-record should be composed according to the default element set name for that database.

A target system must always recognize the character string "F" as an element set name to mean "full record". When a "full record" is requested, the target returns all of the elements stored in its database for the requested record. For a given record, the set of elements included in a "full record" in one database might not be the same set of elements included in a "full record" in another database. (The target may use "F" as the default element set name.)

3.2.2.1.5 Preferred-record-syntax The parameter Preferred-record-syntax identifies the preferred abstract syntax for the records to be returned, from among the set of abstract syntaxes for records for which presentation contexts are currently established for this application association. If the target cannot supply the information in the requested abstract syntax, it will supply it in one of the other abstract syntaxes from the established set.

3.2.2.1.6 Small-set-upper-bound, Large-set-lower-bound, and Medium-set-present-number The result set is considered a "small-set", "medium-set", or "large-set", depending on the values of parameters Small-set-upper-bound and Large-set-lower-bound of the Search request, and Result-count of the Search response (see section 3.2.2.1.8). The result set is a small-set if Result-count is not greater than small-set-upper-bound. The result set is a large-set if Result-count is larger than or equal to Large-set-lower-bound. Otherwise, the result set is a medium-set.

If the query results in a small-set, all database records identified by the result set are to be returned to the origin (subject to possible message size constraints). If the query results in a large-set, no database records are to be returned. If the query results in a medium-set, the maximum number of database records to be returned is specified by Medium-set-present-number.

Notes:

- (1) The result set may be a medium-set only when Result-count is greater than small-set-upper-

bound and less than Large-set-lower-bound, and this can occur only if Large-set-lower-bound is at least 2 greater than Small-set-lower-bound; i.e., the result set cannot be a medium-set if Large-set-lower-bound exceeds Small-set-upper-bound by one. For example, if Large-set-lower-bound is 11 and Small-set-upper-bound is 10, the intent is "if 10 or less records are found return them all, otherwise do not return any", and medium-set-present-number would not apply.

- (2) Small-set-upper-bound may be zero. Large-set-lower-bound must be greater than Small-set-upper-bound.
- (3) If the origin does not want any records returned in the response regardless of the value of Result-count, Larger-set-lower-bound should be set to 1 and Small-set-upper-bound to zero.

3.2.2.1.7 Database/diagnostic-records The target processes the search, creating a result set which identifies a set of database records. It cannot be assumed however that search processing requires physical access to the database records; thus one or more records might not be returnable, but this circumstance might not be recognized until an attempt is made to transfer such a record.

After processing the search, the target attempts to retrieve the first N records identified by the result set, to be included in the Search response (N depends on the search parameters and result-count, as described in 3.2.2.1.6). For each record which cannot be returned, a surrogate diagnostic record is substituted. Thus the parameter Database/diagnostic-records is one of the following:

- N database and/or diagnostic records,
- a number of database and/or diagnostic records, which is less than N because of message size constraints (see 3.3);
- a single non-surrogate diagnostic record indicating that the search cannot be processed, and why it cannot be processed; or
- a single non-surrogate diagnostic record indicating that records cannot be presented, and why not, e.g. "element set name not valid for database".

The order of occurrence of records (database and/or surrogate diagnostic) within the parameter Database/diagnostic-records is according to the order in which they are identified by the result set. Each database or surrogate diagnostic record may optionally be accompanied by the name of the database in

which the record resides. However, the database name must accompany the first database or surrogate diagnostic record being returned, and must accompany any record coming from a database different than its immediate predecessor.

3.2.2.1.8 Result-count and Number-of-records-returned The parameter Result-count is the number of records identified by the result set. If the result set is empty, result-count is zero. The parameter Number-of-records-returned is the total number of records (database and diagnostic) returned in the Search response.

3.2.2.1.9 Next-result-set-position The parameter Next-result-set-position specifies the position within the result set of the next record following those returned (or zero if the last record in the result set is being returned).

3.2.2.1.10 Search-status The parameter Search-status, returned in the response, assumes one of the following two values:

- success - the search completed successfully
- failure - the search did not complete successfully

A value of "success" does not imply that the expected database and/or surrogate diagnostic records are being returned as part of the response (see Present-status, 3.2.2.1.11). Note also, a value of "success" does not imply that any records were located by the search. A value of "failure" does imply that none of the expected database and/or surrogate diagnostic records is being returned. In the latter case, the target returns a single diagnostic record indicating that the search cannot be processed.

3.2.2.1.11 Result-set-status and Present-status These are status descriptors necessary to disambiguate certain situations that can occur during search and present operations.

Result-set-status occurs if and only if the value of Search-status is "failure", and its value is one of the following:

- subset - Partial, valid results available.
- interim - Partial results available, not necessarily valid.
- none - No results available (result set is empty).

Present-status occurs if and only if the value of Search-status is "success", and its value is one of the following:

- success - All of the expected database (or surrogate diagnostic) records are available.
- partial-1 - Not all of the expected records can be returned because the request was terminated by access-control.
- partial-2 - Not all of the expected records can be returned because the request was terminated by maximum message size.
- partial-3 - Not all of the expected records can be returned because the request was terminated by resource-control at origin.
- partial-4 - Not all of the expected records can be returned because the request was terminated by resource-control at target.
- failure - None of the expected database (or surrogate diagnostic) records can be returned. A single diagnostic is returned, which is not a surrogate for a database record.

3.2.2.1.12 Reference-id See section 3.4

3.2.3 Retrieval Facility

The Retrieval facility enables the origin to retrieve a copy of records according to position within a result set maintained by the target.

3.2.3.1 Present Service

The Present service allows the origin system to retrieve records from a specified result set. Records are referenced by relative position within the result set. The origin specifies a range of records to be retrieved and may follow with subsequent requests specifying different ranges. For example, the origin may retrieve records one through five and follow with a request for records four through six.

3.2.3.1.1 Number-of-records-requested and Result-set-start-position The origin requests the return of N records beginning at record M, from the result set, where $N = \text{Number-of-records-requested}$ and $M = \text{Result-set-start-position}$ (and N is not greater than $(\text{Result-count} - M) + 1$).

3.2.3.1.2 Result-set-id Result-set-id specifies the result set from which records are to be retrieved. It

Parameter	Origin Request	Target Response
Number-of-records-requested	x	
Result-set-start-position	x	
Result-set-id	x	
Element-set-names	x (optional)	
Preferred-record-syntax	x (optional)	
Database/diagnostic-records		x (if appl.)
Number-of-records-returned		x
Next-result-set-position		x
Present-status		x
Reference-id		x (if appl.)

Table 3: Parameters of the Present Service

is the result set created by a previous Search request for which the value of the parameter Result-set-name matches the value of Result-set-id.

3.2.3.1.3 Element-set-names The parameter Element-set-names describes the preferred composition of the records expected in the Present response. See section 3.2.2.1.4.

3.2.3.1.4 Preferred-record-syntax See section 3.2.2.1.5.

3.2.3.1.5 Database/diagnostic-records The parameter Database/diagnostic-records returned by the target consists of one of the following:

- N database and/or diagnostic records, where $N = \text{Number-of-records-requested}$,
- a number of database and/or diagnostic records, which is less than N (reason specified by Present-status), or
- a single diagnostic record indicating that the request cannot be processed, and why it cannot be processed.

The order of occurrence of records (database and/or surrogate diagnostic) within the parameter

Database/diagnostic-records is according to the order in which they are identified by the result set. Each database and/or surrogate diagnostic record may optionally be accompanied by the name of the database in which the record resides. However, the database name must accompany the first record being returned, and must accompany any record coming from a database different than its immediate predecessor.

3.2.3.1.6 Number-of-records-returned and Next-result-set-position The parameter Number-of-records-returned is the total number of database and diagnostic records returned. Next-result-set-position is the position within the result set of the next record following the last database or surrogate diagnostic record being returned (or zero, if the last database or surrogate diagnostic record in the result set is being returned).

3.2.3.1.7 Present-status Present-status is mandatory in a Present response and its values are the same as those listed for Present-status in 3.2.2.1.11.

3.2.3.1.8 Reference-id See section 3.4.

3.2.4 Result-set-delete Facility

The Result-set-delete facility enables an origin to instruct a target system to delete one or more result sets known to the target. The target responds with information pertaining to the result of the operation.

3.2.4.1 Delete Service

The Delete service enables an origin system to send a Delete request to the target. The origin may request deletion of specified result sets held by the target or all result sets currently on the target created during this association. The target responds by reporting the status of the requested delete operation.

3.2.4.1.1 Delete-operation The origin specifies one of the following:

- list - delete one or more specified result sets (see 3.2.4.1.2), or
- bulk-delete - delete all result sets currently on the target system created during this association.

3.2.4.1.2 Result-set-list If Delete-operation is "list", then the parameter Result-set-list must be present. It specifies a list of result sets to be deleted, which were created by previous Search requests (in

<u>Parameter</u>	<u>Origin Request</u>	<u>Target Response</u>
Delete-operation	x	
Result-set-list	x (if appl.)	
Delete-operation-status		x
Delete-list-statuses		x (if appl.)
Number-not-Deleted		x (if appl.)
Bulk-statuses		x (if appl.)
Delete-msg		x (optional)
Reference-id	x (optional)	x (if appl.)

Table 4: Parameters of the Delete Service

which the value of the parameter Result-set-name matches the value of one of the members of the list). If Delete-operation is "bulk-delete", then the parameter Result-set-list must not be present.

3.2.4.1.3 Delete-operation-status Delete-operation-status is used by the target to report the status of the delete request. It assumes one of the values "success" or "failure-3" through "failure-9" in table 5.

3.2.4.1.4 Delete-list-statuses Delete-list-statuses is present in a Delete response for a list operation. It contains the same Result-set-id(s) contained in the Result-set-list parameter of the corresponding Delete request. Each result-set-id in the Delete-list-statuses parameter is paired with a status. Possible status values are "success" and "failure-1" through "failure-6" in table 5.

3.2.4.1.5 Number-not-deleted and Bulk-statuses These two parameters occur only if Delete-operation is Bulk-delete and if Delete-operation-status = "failure-8". The parameter Number-not-deleted indicates how many result sets were not deleted, and the parameter Bulk-statuses gives individual statuses for those not deleted.

Note, however, that a target system is not obligated to provide statuses for each result set not deleted on a bulk delete. For example, a target system may abort a bulk delete when the first failure to delete a

Status	Description
success	Result set(s) deleted.
failure-1	Result set did not exist.
failure-2	Result set previously unilaterally deleted by target.
failure-3	System problem at target (optional text message may be included in the Delete-msg parameter).
failure-4	Access-control failure: the delete request caused the target system to issue an Access-control request which the origin system failed to satisfy, or the origin could not accept an Access-control request.
failure-5	Request terminated by origin system through resource control.
failure-6	Access terminated by target system due to resource constraints.
failure-7	Bulk delete of result sets not supported by target.
failure-8	Not all result sets deleted (on a bulk-delete request) (see 3.2.4.1.5).
Failure-9	Not all requested result sets deleted (on a list request).
Note: failure-7 and failure-8 can occur only if Delete-operation is Bulk-delete.	

Table 5: Delete statuses

result set that is part of the bulk delete fails; in this case only a single status might be included in the parameter Bulk-statuses.

If a bulk delete results in more statuses than can fit into a single Delete-response message, the target system may discard those which do not fit.

3.2.4.1.6 Delete-msg Delete-msg, if present, contains an optional text message.

3.2.4.1.7 Reference-id See section 3.4.

3.2.5 Access Control Facility

The Access-control facility allows a target system to challenge an origin system during execution of an Init, Search, Present, Delete, or Resource-report operation.

3.2.5.1 Access-control Service

An origin system must be prepared to accept and respond to one or more Access-control requests while an Init, Search, Present, Delete, or Resource-report request is being executed by the target system (unless the parameter Options of the Init request which initiated the connection did not include Access-control; see 3.2.1.1.2). For example, after sending a Search request, the origin must be prepared to receive an Access-control request, respond with an Access-control response, then later receive another Access-control request, etc., before receiving a Search response.

Once the origin has responded, the operation proceeds as if the challenge has never taken place. If the origin system fails to respond correctly to the challenge during a Search, Present, Delete, or Resource-report request, then the respective response may indicate that the operation was terminated due to an Access-control failure. (Note: During a Search or Present operation, while the target is preparing records for presentation, it might send an Access Control request pertaining to a particular record. If the origin fails to respond correctly to the challenge, the target may simply substitute a surrogate diagnostic: "security challenge failed; record not included".) If the origin system fails to respond correctly to the challenge during an Init request, the target may set the Result parameter to "reject" and may (optionally) supply such an indication in the User-information-field of the Init response.

The Access-control request/response mechanism can be used to support password challenges, public key cryptosystems, algorithmic authentication, etc. The specific content of the Access-control request and response parameters are outside the scope of this standard.

3.2.5.1.1 Security-challenge and Security-challenge-response The contents of these two parameters are outside the scope of this standard and must be established by prior agreement between a given target/origin system pair.

3.2.5.1.2 Reference-id See section 3.4.

<u>Parameter</u>	<u>Target Request</u>	<u>Origin Response</u>
Security-challenge	x	
Security-challenge-response		x
Reference-id	x (if appl.)	x (if appl.)

Table 6: Parameters of the Access Control Service

3.2.6 Accounting/Resource Control Facility

The Accounting/Resource Control facility consists of the Resource-control service (initiated by the target during a Search, Present, or Delete operation), the Trigger-resource-control service (initiated by the origin during a Search, Present, or Delete operation), and the Resource-report service (initiated by the origin when no operation is pending).

The Resource-control service permits the target system to send a Resource-report, notifying the origin system when either actual or predicted resource consumption will exceed agreed upon limits (or limits built into the target system), and to obtain the origin system's consent to continue the operation. In addition, the target system can inform the origin system about the current status of a result set being generated on the target in response to a Search request, and indicate information about the status of the current request to the origin.

The Trigger-resource-control service permits the origin system to request that the target initiate the Resource-control service, or cancel the current operation.

When no operation is pending, the Resource-report service permits the origin system to request that the target send a Resource-report.

3.2.6.1 Resource-control Service

A target system may issue a Resource-control request following receipt of an Init, Search, Present, or Delete request, prior to issuing the corresponding Init, Search, Present, or Delete response.

The target indicates whether a response to the request is required:

- If the target indicates that a response is required, the origin system must issue a Resource-control response. The target awaits the Resource-control response, and subsequently issues the Init, Search, Present, or Delete response, after processing of the operation is concluded.
- If the target indicates that a response is not required, the origin system must not issue a Resource-control response, and the target subsequently issues the Init, Search, Present, or Delete response, after processing of the operation is concluded.

An origin should be prepared to receive, and (conditionally) respond to, multiple Resource-control requests during the execution of an Init, Search, Present, or Delete request.

If the origin responds to a Resource-control request with a Resource-control response saying to terminate the operation, it can expect to receive an Init, Search, Present, or Delete response. This response might indicate that the operation was terminated at origin request. However, the response might alternately indicate that the operation completed, since the operation at the target system may continue to execute and subsequently complete before the Resource-control response reaches the target system.

<u>Parameter</u>	<u>Target Request</u>	<u>Origin Response</u>
Resource-report	x (opt.)	
Partial-results-available	x (if appl.)	
Suspended-flag	x (if appl.)	
Response-required	x	
Triggered-request-flag	x (opt.)	
Continue-flag		x
Result-set-wanted		x (if appl.)
Reference-id	x (if appl.)	x (if appl.)

Table 7: Parameters of the Resource Control Service

3.2.6.1.1 Resource-report Resource-report may be used to convey information about the current and estimated resource consumption at the target system. The format of Resource-report bib-1 is defined in Appendix F.

3.2.6.1.2 Partial-results-available The target indicates the status of the result set via the flag Partial-results-available, whose value is one of the following:

- subset - Partial, valid results available.
- interim - Partial results available, not necessary valid.
- none - No results available.

This parameter is meaningful only during a search operation. If its value is "subset" or "interim", then the target will accept subsequent Present requests if the current request is terminated by the Resource-control response, and if the value of the parameter Result-set-wanted is "on".

If the value of Partial-results-available is "none" then the target need not accept subsequent Present requests in the event that the request is terminated by the Resource-control response.

Note that if Suspended-flag is off, the partial results available situation may change since processing continues on the search. In all cases, the values of Search-status and Result-set-status in the Search response should be treated as the authoritative information.

3.2.6.1.3 Suspended-flag The target system indicates whether or not processing of the command has been suspended pending the Resource-control response. This flag occurs if and only if the value of Response-required is "yes".

3.2.6.1.4 Response-required The target system indicates whether or not a response (from the origin) to this request is required.

3.6.2.1.5 Triggered-request-flag The target system may optionally indicate whether or not this request resulted from a Trigger-resource-control request from the origin.

3.2.6.1.6 Continue-flag The origin indicates to the target whether or not to continue processing.

3.2.6.1.7 Result-set-wanted This flag is meaningful only:

- during a Search request,
- when the value of Partial-results-available is "subset" or "interim", and
- when the value of the parameter Continue-flag is "do not continue".

If the value of the flag is "on", the target will maintain the (possibly partial) result set for subsequent Present requests. If the value of the flag is

"off", the target may delete the result set. A result set status of "none" on the subsequent Search response indicates that the target has discarded the result set. In all cases, the values of Search-status and Result-set-status in the Search response describe the actual decisions made by the target system and the way in which the search terminated.

3.2.6.1.8 Reference-id See section 3.4.

3.2.6.2 Trigger-resource-control Service

An origin system may issue a Trigger-resource-control requests following an Init, Search, Present, or Delete request, prior to receipt of the corresponding response. The request serves as a signal to the target system that the origin wishes the target to:

- a) simply send a Resource-report -- i.e. issue a Resource-control request with Response-required "off";
- b) invoke full resource control -- i.e. issue a Resource-control request with Response-required "on"; or
- c) Cancel the current Init, Search, Present, or Delete operation.

The target is not obligated to take any specific action upon receipt of a Trigger-resource-control request. For the purpose of procedure description, there is no response to the request; if the target wishes to issue a Resource-control request it does so unilaterally. (If the origin issues a Trigger-resource-control request and subsequently receives a Resource-control request, the origin cannot necessarily determine whether the latter resulted from the Trigger-resource-control request. However, the target may include Triggered-request-flag in the Resource-control-request to so indicate.)

If the origin issues a Trigger-resource-control request saying to cancel the command, and if the target honors the request, the origin can expect to receive an Init, Search, Present, or Delete response indicating that the request was terminated at origin request.

Although an origin system may issue a Trigger-resource-control request only prior to receipt of an Init, Search, Present, or Delete response, the target might issue such a response before it receives the Trigger-resource-control request. In that case, the target will ignore the Trigger-resource-control request. Furthermore, the target might receive a Trigger-resource-control request after issuing a Resource-control request, while awaiting a Resource-control response. In that case, again, the target should ignore the Trigger-resource-control request.

(Note that in general, the target may ignore any Trigger-resource-control request.)

Parameter	Origin Request
Requested-operation	x
Preferred-resource-report-format	x (if applicable)
Result-set-wanted	x (if applicable)
Reference-id	x (if applicable)

Table 8: Parameters of the Trigger Resource Control Service

3.2.6.2.1 Requested-operation The origin indicates one of the following:

- resource-report - issue a Resource-control request and set Response-required to "off";
- resource-control - issue a Resource-control request and set Response-required to "on";
- cancel - Terminate the current Init, Search, Present, or Delete operation.

3.2.6.2.2 Preferred-Resource-report-format This parameter is meaningful only when the value of the parameter Requested-operation is "resource-control" or "resource-report". It identifies a resource report format that the origin prefers.

3.2.6.2.3 Result-set-wanted This flag is meaningful only during a Search request and when the value of the parameter Requested-operation is "cancel". If the value of the flag is "on", the origin requests that the target maintain the (possibly partial) result set for subsequent Present requests. See section 3.2.6.1.7.

3.2.6.2.4 Reference-id See section 3.4.

3.2.6.3 Resource-report Service

The origin may issue a Resource-report request following receipt of an Init, Search, Present, or Delete response from the target. The target responds with a Resource-report response.

Note: The Resource-report service is a confirmed service and as such differs from the Trigger-resource-control service, which is non-confirmed. The origin may invoke the Resource-report service

only in the "open" state, that is, following the conclusion of an operation (Init, Search, Present, Delete, or Resource-report) and prior to initiation of a subsequent operation. Therefore a Resource-report request from the origin requires a response from the target. (If the response were not determinable, then the origin would not be certain when to initiate a subsequent operation. In contrast, when the origin issues a Trigger-resource-control request, it awaits an action from the target -- either a Resource-control request or an operation response -- and therefore the lack of a deterministic response does not present sequencing problems.) However, although the target is obliged send a Resource-report response, the target is not obliged to include a resource-report in the Resource-report response.

Parameter	Origin Request	Target Response
Preferred-resource-report-format	x (opt.)	
Resource-report-status		x
Resource-report		x (opt.)
Reference-id	x (opt.)	x (if appl.)

Table 9: Parameters of the Resource Report Service

3.2.6.3.1 Preferred-resource-report-format Identifies a resource report format that the origin prefers.

3.2.6.3.2 Resource-report-status The target supplies one of following status values:

- success - A resource report is included (and in the preferred format, if the parameter Preferred-resource-report-format was included in the request).
- partial - A resource report is included, but not in the preferred format (applies only if the parameter Preferred-resource-report-format was included in the request).
- failure-1 - Target unable to supply resource report.
- failure-2 - Access terminated by target due to resource constraints.
- failure-3 - Access-control failure.
- failure-4 - Unspecified failure.

3.2.6.3.3 Resource-report See 3.2.6.1.1.

3.2.6.3.4 Reference-id See section 3.4.

3.2.7 Termination Facility

The Termination Facility allows either:

- (1) an origin or target to initiate abrupt termination of the association via the IR-abort service element, or
- (2) an origin to initiate graceful termination via the IR-release service.

The IR-abort and IR-release services map directly onto the A-ABORT and A-RELEASE services of the association control service element.

3.2.7.1 IR-abort Service

Either the origin or target may at any time either send or receive an IR-abort request, and consider the application association terminated.

3.2.7.2 IR-Release Service

The origin may invoke an IR-release request following receipt of an Init, Search, Present, Delete, or Resource-report response. It should then await receipt of an IR-Release response, and then consider the association terminated. Alternately, it might receive an IR-abort request from the target, in which case it should consider the application association terminated.

The target may receive an IR-release request after sending an Init, Search, Present, Delete, or Resource-report response. It should then send an IR-release response, and consider the association terminated.

3.3 Message Size Limitations

For both the Search and Present services, it is possible that the target will not be able to return the number of database records requested because of message size limitations. In that case, the target is responsible for packing as many records as possible into the response message. (Note: A response message always contains an integral number of records; a record never spans response messages.)

Illustration

Assume that the target is attempting to transmit records in result set positions 1 through 10 (in this section, the term "record" means "database or surro-

gate diagnostic record", unless "diagnostic record" or "database record" is specified). Assume further that:

- records in position 1 through 6 fit in the response message, such that the sum of the sizes of the records (not including any protocol control information) does not exceed Preferred-message-size; but,
- the database record in position 7 will not fit in the message along with records in position 1 through 6 without the resulting sum of the message sizes exceeding Preferred-message-size.

The size of the database record in position 7:

- (a) does not exceed Preferred-message-size, or
- (b) exceeds Preferred-message-size, but does not exceed Maximum-record-size, or
- (c) exceeds Maximum-record-size.

In case (a), the target returns records in position 1 through 6. In case (b), except as noted below (see "Exception"), the target substitutes a diagnostic record for the database record in position 7, indicating that the record exceeds Preferred-message-size. In case (c) the target substitutes a diagnostic record for the database record in position 7, indicating that the record exceeds Maximum-record-size. (If Maximum-record-size equals Preferred-message-size then there is no distinction between the meaning of the two diagnostics.)

In case (b) or (c) if the diagnostic record will not fit along with the records in position 1 through 6, the target returns the records in position 1 through 6. (Preferred-message-size must always be large enough to contain any diagnostic record; thus a subsequent present request beginning with the record in position 7 will retrieve the diagnostic.) Otherwise, the target inserts the diagnostic record and proceeds to attempt to fit records in positions 8 through 10 into the response message.

Exception

If a Present request specifies a single record (i.e. Number-of-records-requested equals 1) then if the size of that record exceeds Preferred-message-size, but does not exceed Maximum-record-size, the target will return that single database record in the Present response. Note that this exception applies only to a Present request and not to a Search request.

Thus in case (b), the origin may subsequently request the database record in position 7, by issuing a Present request in which that record is the only record requested.

Note that the purpose of this distinction between Preferred-message-size and Maximum-record-size is to allow the transfer of normal length records to proceed in a routine fashion, with convenient buffer sizes, but to also provide for the transfer of an occasional exceptionally large database record without requiring the origin to continually allocate and hold local buffer space for worst-case records. Note also that this intended purpose is defeated if the origin routinely requests a single record.

3.4 Reference-id

The parameter Reference-id is optional in an Init, Search, Present, Delete, and Resource-report request. If supplied,

- it must be echoed by the target in the respective response;
- it must be echoed in both the request and response of any intermediate Access-control or Resource-control request/response sequences; and
- it must be echoed by the origin in any intermediate Trigger-resource-control request.

If Reference-id is not supplied in an Init, Search, Present, Delete, or Resource-report request, then it is not to appear in the respective response, nor in either the request or response of any intermediate Access-control or Resource-control request/response sequence, nor in any intermediate Trigger-resource-control request.

The service does not assume any relationship between the Reference-id used in an Init, Search, Present, Delete, or Resource-report request and the Reference-id used in any other Init, Search, Present, Delete, or Resource-report request.

The purpose of this parameter is to facilitate the grouping of events by the origin system. This standard does not specify its contents. Reference-ids are always assigned by the origin and have meaning only within the origin system. Since there are no semantics attributed to this parameter, it has no implied datatype and can only be described as transparent binary data. It is therefore described as ASN.1 type OCTETSTRING.

4. Protocol Specification

This version of the ANSI Z39.50-1992 Information Retrieval protocol is version 2.

Note: For the purpose of interworking with version 1 of the Search and Retrieve Protocol (ISO 10163), Version 2 of Z39.50 is assumed identical to version 1 of Z39.50; thus implementations which support version 2 automatically support version 1. Implementations that intend to interwork with SR implementations as well as other Z39.50 implementations should indicate support for both versions 1 and 2 (Version 1 of ANSI Z39.50-1992 should not be confused with ANSI Z39.50-1988).

The Information Retrieval application protocol specifies the formats and procedures governing the transfer of information between peer information retrieval applications. A unit of information, passed between two peer applications is called an "application protocol data unit" or APDU.

4.1 Abstract Syntax of the Information Retrieval Protocol

The Information Retrieval protocol data units are complex data types. The transfer syntax of these data types is negotiated by the presentation service provider. The definitions in this section specify the component data elements of the protocol data units and the Type-1 query.

4.1.1 ASN.1 Specification of the APDUs

This section describes the abstract syntax of the Z39.50 APDUs, using the ASN.1 notation defined in ISO 8824. The comments included within the ASN.1 specification constitute part of the standard.

IR DEFINITIONS ::=

BEGIN -- ANSI Z39.50 version 2 - July 1, 1992

PDU ::= CHOICE{

initRequest	[20]	IMPLICIT InitializeRequest,
initResponse	[21]	IMPLICIT InitializeResponse,
searchRequest	[22]	IMPLICIT SearchRequest,
searchResponse	[23]	IMPLICIT SearchResponse,
presentRequest	[24]	IMPLICIT PresentRequest,
presentResponse	[25]	IMPLICIT PresentResponse,
deleteResultSetRequest	[26]	IMPLICIT DeleteResultSetRequest,
deleteResultSetResponse	[27]	IMPLICIT DeleteResultSetResponse,
accessControlRequest	[28]	IMPLICIT AccessControlRequest,
accessControlResponse	[29]	IMPLICIT AccessControlResponse,
resourceControlRequest	[30]	IMPLICIT ResourceControlRequest,
resourceControlResponse	[31]	IMPLICIT ResourceControlResponse,
triggerResourceControlRequest	[32]	IMPLICIT TriggerResourceControlRequest,
resourceReportRequest	[33]	IMPLICIT ResourceReportRequest,
resourceReportResponse	[34]	IMPLICIT ResourceReportResponse}

-- new APDUs can be added in the future at the end of this list.

InitializeRequest ::= SEQUENCE{

referenceId	ReferenceId OPTIONAL,
protocolVersion	ProtocolVersion,
-- proposed version of the protocol to be used (see below).	
options	Options,
-- proposed set of services to be used	
preferredMessageSize	PreferredMessageSize,
-- origin proposal for the size of large messages (where message size is	
-- the sum of sizes, in bytes, of the records in an APDU) the target	
-- should normally use when presenting groups of records; however, the	
-- first record in a response is permitted to cause the message to	
-- exceed this size (see also maximumRecordSize below).	
maximumRecordSize	MaximumRecordSize,
-- origin proposal for maximum record size (number of bytes).	
-- Value must be greater than or equal to	
-- preferredMessageSize.	
idAuthentication	[7] ANY OPTIONAL,
-- information as required by the target to access the	
-- responding IRPM; syntax of this parameter to be	
-- defined by the target prior to communication.	
implementationId	ImplementationId OPTIONAL,
implementationName	ImplementationName OPTIONAL,
implementationVersion	ImplementationVersion OPTIONAL,
userInfoField	UserInfoField OPTIONAL}

InitializeResponse ::= SEQUENCE{

referenceId	ReferenceId OPTIONAL,
protocolVersion	ProtocolVersion,
options	Options,
preferredMessageSize	PreferredMessageSize,

-- target decision on maximum APDU size (see description under
-- InitializationRequest definition). Value is allowed to be different
-- from what origin proposed in the Initialization Request; if origin
-- does not agree on target values, it may abort the connection.

-- (InitializeResponse continued next page)

-- (InitializeResponse continued)

maximumRecordSize	MaximumRecordSize,
-- target decision on the maximum record size	
result	[12] IMPLICIT BOOLEAN,
-- result of the processing of the request at the target.	
-- reject = FALSE; Accept = TRUE.	
implementationId	ImplementationId OPTIONAL,
implementationName	ImplementationName OPTIONAL,
implementationVersion	ImplementationVersion OPTIONAL,
userInformationField	UserInformationField OPTIONAL}

-- Auxiliary definitions for Initialization APDUs

ProtocolVersion ::= [3] IMPLICIT BIT STRING

- represents a string of Boolean values, each value representing
- a version. The first value set to 1 indicates version 1 is available,
- the second value set to 1 indicates version 2 is available, and so on.
- Values higher than the highest known version should be ignored. Both the
- Initialize and Initialize Response APDUs include a value string
- corresponding to the supported versions. The highest common version is
- selected for use. If there are no versions in common, the Initialize
- Response APDU should indicate a value of "reject" for the parameter
- Result. Note: Versions 1 and 2 are identical. Systems supporting
- version 2 should indicate support for version 1 as well,
- for interoperability with systems that indicate support for
- version 1 only (e.g. ISO 10163 implementations).

Options ::= [4] IMPLICIT BIT STRING

{search	(0),
present	(1),
delSet	(2),
resourceReport	(3),
triggerResourceCtrl	(4),
resourceCtrl	(5),
accessCtrl	(6)}

- In InitializeRequest, for bits 0 through 4, ON indicates initiator does
- request use of service; for bits 5 and 6, ON indicates initiator will
- support service. For InitializeResponse, for bit 0 through 4, ON indicates
- target will support service; for bits 5 and 6, ON indicates target requests
- service. For extensibility of the protocol, additional bits set should
- not be considered to be an error on received InitializeRequest.

PreferredMessageSize	::= [5]	IMPLICIT INTEGER
MaximumRecordSize	::= [6]	IMPLICIT INTEGER

ImplementationId ::= [110] IMPLICIT VisibleString

-- a unique identifier for the origin or target implementation

ImplementationName ::= [111] IMPLICIT VisibleString

-- a descriptive name for the origin or target implementation

ImplementationVersion ::= [112] IMPLICIT VisibleString

-- descriptive name for the origin or target impl. version

- these three implementation parameters are provided solely for
- the convenience of implementors needing to distinguish implemen-
- tations. They shall not be the subject of conformance tests.

UserInformationField ::= [11] EXTERNAL

-- additional information, not defined in this standard.

-- end auxiliary definitions for initialization APDUs

SearchRequest ::= SEQUENCE{

referenceId **ReferenceId** OPTIONAL,
smallSetUpperBound **[13] IMPLICIT INTEGER**,
-- if the number of hits is less than or equal to this number, all records are
-- to be returned in the SearchResponse (within the limits given by the
-- negotiated preferred-MessageSize and maximumRecordSize), composed as
-- specified by the smallSetElementSetNames parameter below. May be zero.
largeSetLowerBound **[14] IMPLICIT INTEGER**,
-- if the number of hits is equal to or greater than this number, no records are
-- to be returned. LargeSetLowerBound must be greater than smallSetUpperBound.
mediumSetPresentNumber **[15] IMPLICIT INTEGER**,
-- maximum number records to be returned in the SearchResponse if the number
-- of hits is between largeSetLowerBound and smallSetUpperBound (again
-- within the limits given by PreferredMessageSize and MaximumRecordSize),
-- composed as specified by mediumSetRecordElementSetNames
replaceIndicator **[16] IMPLICIT BOOLEAN**,
-- origin indicates whether or not to replace a previously created result
-- set with the same name by the one that is constructed in this search.
resultSetName **[17] IMPLICIT VisibleString**,
-- origin proposal for naming of the result set that is constructed if hits
-- are found. If target allows the origin to assign names to result sets,
-- it uses this proposed name. If the target does not support named result
-- sets, it returns an error diagnostic.
databaseNames **[18] IMPLICIT SEQUENCE OF DatabaseName**,
-- database(s) in which the search will be performed.
smallSetElementSetNames **[100] IMPLICIT ElementSetNames** OPTIONAL,
-- origin proposal for composition of the records in the small
-- set (see above under smallSetUpperBound).
mediumSetElementSetNames **[101] IMPLICIT ElementSetNames** OPTIONAL,
-- origin proposal for the composition of the records in the
-- medium set (see above under mediumSetPresentNumber).
preferredRecordSyntax **PreferredRecordSyntax** OPTIONAL,
-- origin proposal for abstract syntax of the database records to be returned
-- in the SearchResponse. Values subject to registration.
query **[21] Query**
-- the search argument (see description below).

Query ::= CHOICE

{type-0 [0] ANY,
type-1 [1] IMPLICIT RPNQuery,
type-2 [2] OCTET STRING,
type-100 [100] OCTET STRING,
type-101 [101] IMPLICIT OCTET STRING}
-- the search argument contained in the SearchRequest. Five types are defined:
-- a) A type-0 query may be used only when the origin and target have an a priori
-- agreement outside of this standard as to form of query acceptable to target.
-- b) type-1 is the Reverse Polish Notation (RPN) query (see below).
-- c) type-2 is the ISO8777 type query, specified in ISO 8777.
-- d) type-100 is the Z39.58 type query, specified in ANSI Z39.58.
-- e) type-101 is the proximity query, specified in Appendix G.

RPNQuery ::= SEQUENCE{

attributeSetId **OBJECT IDENTIFIER**, *-- identifies attribute set*
RPNStructure}

```

RPNStructure ::= CHOICE { [0] Operand,
                           [1] IMPLICIT SEQUENCE {
                               RPNStructure,
                               RPNStructure,
                               Operator }}

Operand ::= CHOICE{
    AttributesPlusTerm,
    ResultsetId}
AttributesPlusTerm ::= [102] IMPLICIT SEQUENCE{
    AttributeList,
    Term}
AttributeList ::= [44] IMPLICIT SEQUENCE OF AttributeElement
Term ::= [45] IMPLICIT OCTET STRING
    -- the type OCTET STRING is used to enable the inclusion of multibyte
    -- character representations which the types CharacterString and
    -- VisibleString might impose on graphic character repertoire.
Operator ::= [46] CHOICE{
    and [0] IMPLICIT NULL,
    or [1] IMPLICIT NULL,
    and-not [2] IMPLICIT NULL}

```

```

AttributeElement ::= SEQUENCE{
    AttributeType,
    AttributeValue}
AttributeType ::= [120] IMPLICIT INTEGER
AttributeValue ::= [121] IMPLICIT INTEGER
    -- AttributeType and AttributeValue interpretation is governed by the
    -- values contained in the definition identified by AttributeSetId

```

```

SearchResponse ::= SEQUENCE{
    referenceId ReferenceId OPTIONAL,
    resultCount [23] IMPLICIT INTEGER,
        -- number of hits resulting from the search.
    numberOfRecordsReturned NumberOfRecordsReturned,
        -- number of records in the search results below.
    nextResultSetPosition NextResultSetPosition,
        -- the ordinal number in the result set of the record appearing directly
        -- after the last record in the search Results below.
    searchStatus [22] IMPLICIT BOOLEAN,
        -- result of the processing of the request at the target IRPM.
        -- success = TRUE; failure = FALSE.
    resultSetStatus [26] IMPLICIT INTEGER{
        subset (1),
        interim (2),
        none (3)} OPTIONAL,
        -- occurs if and only if search-status is FALSE. Indicates the
        -- presence and validity of the result set produced.
    presentStatus PresentStatus OPTIONAL,
        -- occurs if and only if search-status is TRUE. Indicates
        -- presence and validity of records appearing in the search results
    databaseOrDiagnosticRecords Records OPTIONAL}
        -- the records (diagnostic and/or bibliographic) resulting
        -- from the search (see description below).

```

PresentRequest ::= SEQUENCE{

referenceId **ReferenceId OPTIONAL,**
resultSetId **ResultSetId,**
 -- identification of the result set from which to retrieve records
resultSetStartPoint **[30] IMPLICIT INTEGER,**
 -- ordinal number in the result set of the first record to
 -- appear in the present results in the PresentResponse.
numberOfRecordsRequested **[29] IMPLICIT INTEGER,**
 -- specifies the maximum number of records to be returned in the present
 -- results in the PresentResponse (within the limits given by the
 -- negotiated message and record size parameters), composed as specified
 -- by the element set names parameter below.
ElementSetNames **OPTIONAL,**
 -- origin proposal for the composition of the records to be
 -- returned in the in the Present Response
preferredRecordSyntax **PreferredRecordSyntax OPTIONAL}**
 -- origin proposal for abstract syntax of the database records to be returned
 -- in the present results in the Present Response. Values subject
 -- to registration, at present by specification in Appendix E.

PresentResponse ::= SEQUENCE{

referenceId **ReferenceId OPTIONAL,**
numberOfRecordsReturned **NumberOfRecordsReturned,**
 -- number of records returned in the records parameter below.
nextResultSetPosition **NextResultSetPosition,**
 -- ordinal number in the result set of the record
 -- appearing directly after the last record returned
presentStatus **PresentStatus,**
 -- indicates the presence and validity of the records
databaseOrDiagnosticRecords **Records OPTIONAL} -- the presented records**

-- begin auxiliary definitions for search and present APDUs

Records ::= CHOICE{

dataBaseOrSurDiagnostics **[28] IMPLICIT SEQUENCE OF NamePlusRecord,**
nonSurrogateDiagnostic **[130] IMPLICIT DiagRec}**

NamePlusRecord ::= SEQUENCE{

[0] IMPLICIT DatabaseName OPTIONAL,
 -- presence of DatabaseName is conditional. See 3.2.2.1.7 and 3.2.3.1.5.
[1] CHOICE{ databaseRecord **[1] DatabaseRecord,**
 surrogateDiagnostic **[2] DiagRec}}**

DatabaseRecord ::= EXTERNAL

-- the database record syntax is defined outside of this standard.
-- For bibliographic data, USMARC is a prime example.

DiagRec ::= SEQUENCE{

diagnosticSetId **OBJECT IDENTIFIER,**
condition **INTEGER,**
 -- interpretation of condition is governed by values
 -- contained in definition identified by DiagnosticSetId.
addinfo **VisibleString} -- add'l information.**

```

ElementSetNames ::= [19] CHOICE{
    generic          [0] IMPLICIT ElementSetName,
    databaseSpecific [1] IMPLICIT SEQUENCE OF SEQUENCE{
        DatabaseName,
        ElementSetName}}
--
        ElementSetName ::= [103] IMPLICIT VisibleString
-- A target must always recognize the value "F" to mean "full record".
--
-- begin miscellaneous definitions for search and present APDUs
    NumberOfRecordsReturned ::= [24] IMPLICIT INTEGER
    NextResultSetPosition    ::= [25] IMPLICIT INTEGER
    PresentStatus             ::= [27] IMPLICIT INTEGER{
        success              (0),
        partial-1            (1),
        partial-2            (2),
        partial-3            (3),
        partial-4            (4),
        failure               (5)}
    PreferredRecordSyntax    ::= [104] IMPLICIT OBJECT IDENTIFIER
-- end miscellaneous definitions for search and present APDUs
-- end auxiliary definitions for search and present APDUs
--
DeleteResultSetRequest ::= SEQUENCE{
    referenceId          ReferenceId OPTIONAL,
    deleteOperation     [32] IMPLICIT INTEGER{
        list             (0),
        all               (1)},
    resultSetList        SEQUENCE OF ResultsetId OPTIONAL }
-- identification of result sets to be deleted if operation is "list"
--
DeleteResultSetResponse ::= SEQUENCE{
    referenceId          ReferenceId OPTIONAL,
    deleteOperationStatus [0] IMPLICIT DeleteSetStatus,
    -- Reports status for entire delete operation. Values limited to
    -- "success" or "failure-3" through "failure-9". Values of "failure-7"
    -- and "failure-8" may be used only if operation is "all".
    deleteListStatuses  [1] IMPLICIT ListStatuses OPTIONAL,
    -- Must occur if operation is "list". Individual status
    -- values limited to "success" through "failure-6".
    numberNotDeleted     [34] IMPLICIT INTEGER OPTIONAL,
    -- No. sets target failed to delete. Occurs only if operation is "all".
    bulkStatuses         [35] IMPLICIT ListStatuses OPTIONAL,
    -- occurs if and only if DeleteSetStatus equals 8. Individual
    -- statuses limited to "success" through "failure-6"
    deleteMessage        [36] IMPLICIT VisibleString OPTIONAL }
-- textual message concerning the delete operation.
--
-- begin auxiliary definitions for delete
    ListStatuses ::= SEQUENCE OF SEQUENCE{
        ResultsetId,
        DeleteSetStatus}

```

```

DeleteSetStatus ::= [33] IMPLICIT INTEGER{
    success                                (0),
    resultSetDidNotExist                   (1),
    previouslyDeletedByTarget              (2),
    systemProblemAtTarget                  (3),
    accessNotAllowed                       (4),
    resourceControlAtOrigin                (5),
    resourceControlAtTarget                (6),
    bulkDeleteNotSupported                 (7),
    notAllResultSetsDeletedOnBulkDelete   (8),
    notAllRequestedResultSetsDeleted       (9)}
    -- 7 and 8 used only if operation is "all".
-- end auxiliary definitions for delete
--
AccessControlRequest ::= SEQUENCE{
    referenceId          ReferenceId OPTIONAL,
    securityChallenge     [37] SecurityParameter}
--
AccessControlResponse ::= SEQUENCE{
    referenceId          ReferenceId OPTIONAL,
    securityChallengeResponse [38] SecurityParameter}
--
-- begin auxiliary definitions for Access Control
SecurityParameter ::= CHOICE{
    oldway          IMPLICIT OCTET STRING,
                    -- for 1988 compatibility
    newway          EXTERNAL}
-- end auxiliary definitions for Access Control
--
ResourceControlRequest ::= SEQUENCE{
    referenceId          ReferenceId OPTIONAL,
    suspendedFlag        [39] IMPLICIT BOOLEAN OPTIONAL,
    -- "true" = suspended
    resourceReport        [40] ResourceReport OPTIONAL,
    partialResultsAvailable [41] IMPLICIT INTEGER{
        subset          (1),
        interim          (2),
        none             (3)} OPTIONAL,
    responseRequired      [42] IMPLICIT BOOLEAN,
    -- "true" means that the target requires a response
    triggeredRequestFlag   [43] IMPLICIT BOOLEAN OPTIONAL,
    -- "true" means request triggered by a trigger-resource-control request
--
ResourceControlResponse ::= SEQUENCE{
    referenceId          ReferenceId OPTIONAL,
    continueFlag         [44] IMPLICIT BOOLEAN, -- true = continue
    resultSetWanted       [45] IMPLICIT BOOLEAN OPTIONAL,
    -- "true" = "result set wanted", required during a search if
    -- Continue flag is false; otherwise should not occur

```

```

TriggerResourceControlRequest ::= SEQUENCE{
    referenceId          ReferenceId OPTIONAL,
    requestedOperation    [46] IMPLICIT INTEGER{
                        resourceReport      (1),
                        resourceControl     (2),
                        cancel              (3)},
    preferredResourceReportFormat [47] IMPLICIT ResourceReportId OPTIONAL,
    resultSetWanted       [48] IMPLICIT BOOLEAN OPTIONAL}

```

```

ResourceReportRequest ::= SEQUENCE{
    referenceId          ReferenceId OPTIONAL,
    preferredResourceReportFormat [49] IMPLICIT ResourceReportId OPTIONAL}

```

```

ResourceReportResponse ::= SEQUENCE{
    referenceId          ReferenceId OPTIONAL,
    resourceReportStatus [50] IMPLICIT INTEGER{
                        success      (0),
                        partial      (1),
                        failure-1    (2),
                        failure-2    (3),
                        failure-3    (4),
                        failure-4    (5)},
    resourceReport       [51] ResourceReport OPTIONAL}

```

-- Begin auxiliary definitions for resource control

ResourceReport ::= EXTERNAL

ResourceReportId ::= OBJECT IDENTIFIER

-- End auxiliary definitions for resource control

-- begin global auxiliary definitions

ReferenceId ::= [2] IMPLICIT OCTET STRING

-- value provided by the service originator in the Request APDU, target

-- required to send it back unaltered in corresponding response APDU

DatabaseName ::= [105] IMPLICIT VisibleString

ResultSetId ::= [31] IMPLICIT VisibleString

-- end global auxiliary definitions

END - IR

4.2 Protocol Procedures

This section specifies services required by the protocol, the protocol model, state tables, handling of protocol errors, rules for extensibility, and conformance requirements.

4.2.1 Services Required

The Information Retrieval protocol assumes service from the Presentation layer and the association control service element.

4.2.1.1 Service Required from the Presentation Layer

The protocol uses the presentation service as defined in ISO 8822 to provide a presentation connection for communication between two information retrieval applications. The presentation services required are those contained in the presentation kernel functional unit and the session duplex functional unit. The association control service element may have additional requirements for presentation services.

All Information Retrieval protocol data units will be mapped onto the P-Data service. The communication service that supports this protocol is a connection-oriented service using the P-DATA service, defined in ISO 8822 in an established application association, in combination with the ACSE, ISO 8649.

A Z39.50 origin establishes application-associations as necessary with the target with whom it is engaged in Z39.50 activity. The Z39.50 application-service-element (ASE) may then use the P-DATA service defined in ISO 8822 directly to transmit Z39.50 APDUs. This provides a connection-oriented interaction between Z39.50 systems. A single application-association can be used to send a series of Z39.50 APDUs relating to multiple searches. A single system can be engaged in multiple application associations with multiple remote systems simultaneously.

4.2.1.2 Association Control Services

Assumed

The protocol assumes the services of the association control service elements as defined in ISO 8649. The services required are:

- 1) orderly association release, where both sides agree to the release and there is no loss of data in transit (the IR-release service is directly mapped to this service without any Information Retrieval protocol control information), and

- 2) association abort, which allows either origin or target, at any time, to explicitly terminate the association, immediately and unconditionally. Data in transit may be lost (the IR-abort service is directly mapped to this service without any Information Retrieval protocol control information).

It is assumed that prior to APDUs being exchanged the Information Retrieval service user will handle the association control services required to establish an association with an application context encompassing the Information Retrieval service. The application context "basic-Z39.50-ac" is registered in appendix A and defined Appendix B.

4.2.2 Protocol Model

To specify protocol procedure, the abstract, implementation-independent concepts of service-user, service-provider, and service primitive are used.

A service-provider provides a communication path between two service users. A service primitive is an element of interaction between the service-user and the service-provider. There are four types of service primitives: Request, Indication, Response and Confirmation. For a confirmed service initiated by the origin (i.e., for Z39.50: Search, Present, Delete, and Resource-report) they are used as follows:

- 1) Request - A primitive issued by the origin service-user to the service-provider in order to invoke some procedure.
- 2) Indication - A primitive issued by the service-provider to the target service-user to indicate that a procedure has been invoked by its peer.
- 3) Response - A primitive issued by the target service-user to the service-provider at the completion of the procedure previously invoked by an indication.
- 4) Confirmation - A primitive issued by the service-provider to the origin service-user, to complete the procedure previously invoked by a request.

Notes:

- (1) For a confirmed service initiated by the target (i.e., for Z39.50: Access-control and Resource-control) the roles of origin and target are reversed.
- (2) For a non-confirmed service (i.e., for Z39.50: Trigger-resource-control) only the Request and Indication primitives are used.

Primitives are conceptual and their use neither specifies nor precludes any specific implementation of a service. Only primitives that correspond to

some element of the service involving the exchange of information between systems are defined.

From the perspective of the service-user, the service-provider is system-independent. For the exchange of protocol however, a distinction is made between those portions of the service-provider residing on the origin and target systems (respectively, the origin service-provider and the target service-provider). See figure 1. The sequence of interactions for a confirmed service initiated by the origin is:

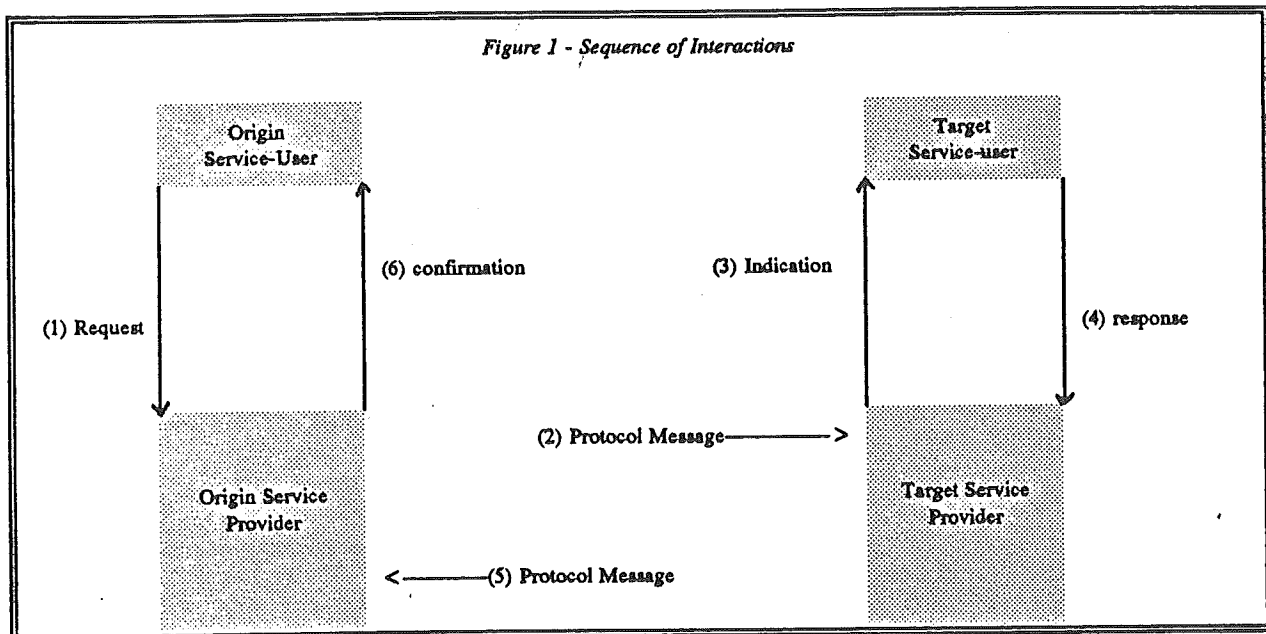
- 1) Request Primitive from origin service-user to service-provider.
- 2) Protocol Message from origin service-provider to target service-provider.
- 3) Indication Primitive from service-provider to target service-user.
- 4) Response Primitive from target service-user to service-provider.
- 5) Protocol Message from target service-provider to origin service-provider.
- 6) Confirmation Primitive from service-provider to origin service-user.

Notes:

- (1) For a confirmed service initiated by the target, the roles of origin and target are reversed.
- (2) For a non-confirmed service, only steps 1 through 3 apply.

The following illustrates the sequence of interactions which occur for a Search operation:

- 1) Search request from origin service-user to service-provider.
- 2) Search APDU (Application Protocol Data Unit) from origin service-provider to target service-provider.
- 3) Search indication from service-provider to target service-user.
- 4) Search response from target service-user to service-provider.
- 5) Search-response APDU from target service-provider to origin service-provider.
- 6) Search confirm from service-provider to origin service-user.



Note: The interfaces between service user and service provider, as represented by steps 1 and 6 for the origin, and by steps 3 and 4 for the target, are described solely to facilitate the specification of protocols. These steps do not represent intersystem communication, and therefore, the means by which they are implemented are not constrained by this specification. In an actual implementation, step 4, for example, might consist of several messages from

the target service user to service provider. On the other hand, both the target service user and service provider could be combined in a single program, in which case steps 3 and 4 might not have any real physical manifestation.

4.2.3 State Tables

This section defines two Information Retrieval Protocol Machines (IRPMs) in terms of state tables.

One state table is defined for the origin (table 10) and one state table is defined for the target (table 11). Each state table shows the inter-relationship between the state of an Information Retrieval association, the incoming events that occur in the protocol, the actions taken, and, finally, the resulting state of the association. The IRPM state table does not constitute a formal definition of the IRPM. It is included to provide a more precise specification of the protocol procedures. The following conventions are used in the state tables.

State Table Cells The intersection of an incoming event (row) and a state (column) forms a cell. A blank cell represents the combination of an incoming event and a state that is not defined for the IRPM. A non blank cell represents an incoming event and state that is defined for the IRPM. Such a cell contains one or more actions, separated by semi-colons (;).

Actions to be Taken by the IRPM The IRPM state tables define the action to be taken by the IRPM in terms of one or more outgoing events (separated by semi-colons) and the resulting state (in parenthesis) of the Information Retrieval association.

Invalid Intersections Blank cells indicate an invalid intersection of an incoming event and state. The state tables define correct operation only. They do not specify actions to be taken in response to incorrect operation (for example, erroneous protocol control information, incorrect protocol control actions, etc.). Such actions are not within the scope of the specification, although implementations must consider them.

4.2.4 Protocol Errors

Any events not listed in the tables of section 4.2.3 are not valid and are considered to be protocol errors. With exceptions specified in section 4.3, incorrectly formatted APDUs or APDU's with invalid data are also considered to be protocol errors. This standard does not specify the actions to be taken upon detection of protocol errors. An application context may contain such a specification.

4.3 Rules for Extensibility

All syntactical errors in received APDUs are considered to be protocol errors except for the following case: Unknown data elements, and unknown options within the Options data element, will be ignored on received Init APDUs.

Table 8: Definition of States

Origin States			Target States		
No.	Name	Description	No.	Name	Description
1	Closed	the origin is awaiting an Init request from the application	1	Closed	the target is awaiting an Init APDU
2	Init sent	the origin has transmitted an Init APDU to the target	2	Init recvd	the target has issued an Init indication
3	Open	the origin is awaiting a Search, Present, Delete, or Resource-report request	3	Open	the target is awaiting a Search, Present, Delete, or Resource-report APDU
4	Search Sent	the origin has transmitted a Search APDU	4	Search recvd	the target has issued a Search indication
5	Prsnt sent	the origin has transmitted a Present APDU	5	Prsnt recvd	the target has issued a Present indication
6	Delete sent	the origin has transmitted a Delete APDU	6	Delete recvd	the target has issued a Delete indication
7	Rsrp sent	the origin has transmitted a Resource-report APDU	7	Rsrp Recvd	the target has issued a Resource-report indication
8	ractrl recvd	the origin has issued a Resource-control indication (response required)	8	ractrl sent	the target has transmitted a Resource-control APDU (response required)
9	Acctrl recvd	the origin has issued an Access-control indication	9	Acctrl sent	the target has transmitted an Access-control APDU
10	Rlease sent	the origin has issued an A_release request	10	Rlease recvd	the target has issued an IR_rel indication.
			11	Reject	the target has transmitted an Init Response APDU (reject)

Table 9a : Events and action and their Abbreviations – Origin

<u>Incoming Event</u>	<u>Abbreviation</u>	<u>Outgoing Action</u>	<u>Abbreviation</u>
Init request	Init req	Init PDU	Init PDU
Init-response PDU	Init resp PDU	Init confirm	Init conf
Search request	Srch req	Search PDU	Srch PDU
Search-response PDU	Srch resp PDU	Search confirm	Srch conf
Present request	Prant req	Present PDU	Prant PDU
Present-response PDU	Prant resp PDU	Present confirm	Prant conf
Delete request	Dlte req	Delete PDU	Dlte PDU
Delete-response PDU	Dlte resp PDU	Delete confirm	Dlte conf
Resource-report request	Rarp req	Resource-report PDU	rarp PDU
Resource-report response PDU	Rarp resp PDU	Resource-report confirm	rarp conf
Trigger-resource-control request	Trigrc	Trigger-resource-control PDU	trigrc PDU
Resource-control PDU (response required)	Rsc PDU (Resp)	Resource-control indication	Rsc ind
Resource-control PDU (response not required)	Rsc PDU (Noresp)		
Resource-control response	Rsc resp	Resource-control-response PDU	Rsc resp PDU
Access-control PDU	Acc PDU	Access-control indication	Acc ind
Access-control response	Acc resp	Access-control-response PDU	Acc resp PDU
A-Abort indication	Aab ind	IR-abort indication	Iab ind
A-P-abort indication	APab ind		
IR-abort request	Iab req	A-abort request	Aab req
IR-release request	Irel req	A-release request	Arel req
A-release confirm	Arel conf	IR-release confirm	Irel conf
		Save current state	stkst
		Restore previously saved state	popst

Table 9b: Events and action and their Abbreviations – Target

<u>Incoming Event</u>	<u>Abbreviation</u>	<u>Outgoing Action</u>	<u>Abbreviation</u>
Init PDU	Init PDU	Init indication	Init ind
Init response	Init resp	Init-response PDU	Init resp PDU
Search PDU	Srch PDU	Search indication	Srch ind
Search response	Srch resp	Search-response PDU	Srch resp PDU
Present PDU	Prant PDU	Present indication	Prant ind
Present response	Prant resp	Present-response PDU	Prant resp PDU
Delete PDU	Dlte PDU	Delete indication	Dlte ind
Delete response	Dlte resp	Delete-response PDU	Dlte resp PDU
Resource-report PDU	Rarp PDU	Resource-report indication	Rarp ind
Resource-report response	Rarp resp	Resource-report-response PDU	Rarp resp PDU
Trigger-resource-control PDU	Trigrc PDU	Trigger-resource-control ind	Trigrc ind
Resource-control Request (response required)	Rsc req (Resp)	Resource-control PDU	Rsc PDU
Resource-control Request (no response required)	Rsc req (Noresp)		
Resource-control-response PDU	Rsc resp PDU	Resource-control confirm	Rsc conf
Access-control Request	Acc req	Access-control PDU	Acc PDU
Access-control-response PDU	Acc resp PDU	Access-control confirm	Acc conf
A-abort indication	Aab ind	IR-abort indication	Iab ind
A-P-abort indication	APab ind		
IR-abort request	Iab req	A-abort request	Aab req
A-release indication	Arel ind	IR-release indication	Irel ind
IR-Release response	Irel resp	A-release response	Arel resp
		Save current state	stkst
		Restore previously saved state	popst

Table 10a: State Table for Origin – Part 1								
Event	State	closed 1	Init sent 2	Open 3	Search sent 4	Prant sent 5	Delete sent 6	Rarp Sent 7
Init req		Init PDU (2)						
Init resp PDU (ACCEPT)			Init conf+ (3)					
Init resp PDU (REJECT)			Init conf-; Arel req (10)					
Srch req				Srch PDU (4)				
Srch resp PDU					Srch conf (3)			
Prant req				Prant PDU (5)				
Prant resp PDU						Prant conf (3)		
Dlte req				Dlte PDU (6)				
Dlte resp PDU							Dlte conf (3)	
Rsrp req				Rarp PDU (7)				
Rsrp resp PDU								Rarp conf (3)
Trigre req			Trigre PDU (2)		Trigre PDU (4)	Trigre PDU (5)	Trigre PDU (6)	

Table 10b: State Table for Origin – Part 2									
State	Init sent	Open	Search sent	Prsnt sent	Delete sent	Rsrp sent	Rscrl recvd	Acctrl recvd	Rlease sent
Event	2	3	4	5	6	7	8	9	10
Rsc PDU (Resp)	Rsc ind; stkst (8)		Rsc ind; stkst (8)	Rsc ind; stkst (8)	Rsc ind; stkst (8)				
Rsc PDU (Noresp)	Rsc ind (2)		Rsc ind (4)	Rsc ind (5)	Rsc ind (6)				
Rsc resp							Rsc resp PDU; popst		
Acc PDU	Acc ind; stkst (9)		Acc ind; stkst (9)	Acc ind; stkst (9)	Acc ind; stkst (9)	Acc ind; stkst (9)			
Acc resp								Acc resp PDU; popst	
Aab ind	lab ind (1)	lab ind (1)	lab ind (1)	lab ind (1)	lab ind (1)	lab ind (1)	lab ind (1)	lab ind (1)	lab ind (1)
Apab ind	lab ind (1)	lab ind (1)	lab ind (1)	lab ind (1)	lab ind (1)	lab ind (1)	lab ind (1)	lab ind (1)	lab ind (1)
lab req	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)
Irel req		Arel req (10)							
Arel conf									Irel conf (1)

Table 11a: State Table for Target – Part 1						
State Event	Closed 1	Init recvd 2	Open 3	Search recvd 4	Prant recvd 5	Delete recvd 6
Init PDU	Init ind (2)					
Init resp (ACCEPT)		Init resp PDU(+) (3)				
Init resp (REJECT)		Init resp PDU(-) (11)				
Srch PDU			Srch ind (4)			
Srch resp				Srch resp PDU (3)		
Prant PDU			Prant ind (5)			
Prant resp					Prant resp PDU (3)	
Dlte PDU			Dlte ind (6)			
Dlte resp						Dlte resp PDU (3)

Table 11b: State table for Target – Part 2										
State Event	Init recvd 2	Open 3	Search recvd 4	Prant recvd 5	Delete recvd 6	Rsrp recvd 7	Ractrl sent 8	Acctrl sent 9	Rlease recvd 10	Reject 11
Rsrp PDU		Rsrp ind (7)								
Rsrp resp						Rsrp resp PDU (3)				
Trigrc PDU	Trigrc ind (2)	ignore (3)	Trigrc ind (4)	Trigrc ind (5)	Trigrc ind (6)		ignore (8)	ignore (9)		ignore (11)
Rsc req (Resp)	Rsc PDU; stkst (8)		Rsc PDU; stkst (8)	Rsc PDU; stkst (8)	Rsc PDU; stkst (8)					
Rsc req (Norep)	Rsc PDU (2)		Rsc PDU (4)	Rsc PDU (5)	Rsc PDU (6)					
Rsc resp PDU							Rsc conf; popst			
Acc req	Acc PDU; stkst (9)		Acc PDU; stkst (9)	Acc PDU; stkst (9)	Acc PDU; stkst (9)	Acc PDU; stkst (9)				
Acc resp PDU								Acc conf; popst		
Arel ind		Irel ind (10)								Irel ind (10)
Irel resp									Arel resp (1)	
Aab ind	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)
Apab ind	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)	Iab ind (1)
Iab req	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)	Aab req (1)

4.4 Conformance

A system claiming to implement the procedures in this standard shall comply with the requirements in sections 4.4.1, 4.4.2, and 4.4.3.

4.4.1 Static Requirements

The system shall:

- a) act in the role of an origin (by sending Init, Search, and Present APDUs and receiving Init-response, Search-response and Present-response APDUs), or target (by responding properly to Init, Search, and Present APDUs with appropriate Init-response, Search-response and Present-response APDUs), or both; and,
- b) support the syntax in section 4.1, and
- c) support the Type-1 Query.

4.4.2 Dynamic Requirements

The system shall exhibit external behavior consistent with having implemented:

- a) an Information Retrieval ASE which follows all the procedures specified in sections 4.1.1, 4.2, and 4.3;
- b) the mapping onto the Association Control Service and Presentation Service (see 4.2.1);
- c) assignment of values to APDU data elements according to the procedures described in section 3;
- d) encoding of APDUs by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the abstract syntax defined in section 4.1.1; and
- e) the Type-1 query whose abstract syntax is defined in section 4.1.1 and whose structure and rules for evaluation are described in section 3.2.2.1.1.1.

4.4.3 Statement Requirements

Each implementation must provide a Protocol Implementation Conformance Statement (PICS).

1. The following shall be stated by the PICS:

- a) whether the system is capable of acting in the role of origin,
- b) whether the system is capable of acting in the role of target,
- c) that the system supports versions 1 and 2 of the Z39.50-1992 protocol.

2. If the system claims the capability of acting in the role of origin the PICS shall state whether the system:

- a) accepts Access-control APDUs and sends Access-control-response APDUs,

- b) accepts Resource-control APDUs and sends Resource-control-response APDUs,
- c) sends Resource-report APDUs and accepts Resource-report response APDUs,
- d) sends Trigger-resource-control APDUs,
- e) sends Delete APDUs and accepts Delete-response APDUs,
- f) sends Search and Present APDUs specifying named result sets other than "default".

3. If the system claims the capability of acting in the role of target, the PICS shall state whether the system:

- a) sends Access-control APDUs and accepts Access-control-response APDUs,
- b) sends Resource-control APDUs and accepts Resource-control-response APDUs,
- c) accepts Resource-report APDUs and sends Resource-report response APDUs,
- d) accepts Trigger-resource-control APDUs,
- e) accepts Delete APDUs and sends Delete-response APDUs,
- f) accepts Search and Present APDUs specifying named result sets other than "default",
- g) unilaterally deletes result sets.

4. The PICS shall state to what extent result sets may be specified as operands in a Type-1 query:

- a) whether named result sets in general, or only the default result set, may be used as an operand in a Type-1 query,
- b) whether result sets may be specified only as the first operand in a Type-1 query, or may be specified as any operand,
- c) with which operators (AND, OR, AND-NOT) result sets may be used as operands.

5. The PICS shall state to what extent element set names are supported in Search and Present APDUs:

- a) whether the parameter Element-set-names is supported,
- b) if the parameter Element-set-names is supported, whether database names corresponding to element set names may be specified, or only a single element set name and no corresponding database name may be specified.

6. The PICS shall state:

- a) for each optional parameter in each APDU, whether or not the parameter is supported;
- b) which encoding rules are supported;
- c) the maximum number of database names which may be specified in a Search APDU;
- d) which attribute sets are supported.

Note: A PICS proforma will be standardized in the future.

APPENDIX A: Object Identifiers Assigned in This Standard

This appendix is part of the standard.

The following object identifier value has been assigned to this standard, ANSI-standard-Z39.50:

{ iso (1) member-body (2) US (840)
ANSI-standard-Z39.50 (10003)}

This standard assigns the following values at the level immediately subordinate to ANSI-standard-Z39.50:

- 1 = application context
- 2 = abstract syntax
- 3 = attribute set
- 4 = diagnostic set
- 5 = abstract record syntax
- 6 = transfer syntax for non-bibliographic records
- 7 = resource report format

A.1 Application Context

This standard assigns the following object identifier for the application-context-definition 'basic-Z39.50-ac', contained in Appendix B:

{ iso member-body US ANSI-standard-Z39.50
application-context (1) basic-Z39.50-ac (1) }

A.2 Abstract Syntax

This standard assigns the following object identifier for the ASN.1 module, contained in section 4.1, defining the Z39.50 APDUs:

{ iso member-body US ANSI-standard-Z39.50
abstract-syntax (2) Z39.50-apdus (1) }

Note: No object identifier is assigned by this standard for any transfer syntax for apdus. The transfer syntax results from the application of the ASN.1 Basic Encoding Rules (ISO 8825). For the

purposes of Presentation Context negotiation, this syntax is identified by a pair of object identifiers, one for the abstract-syntax (Z39.50-apdus) and one for the basic encoding rules:

{ joint-iso-ccitt asn1 (1) basic-encoding (1) }

A.3 Attribute set

This standard assigns the following object identifier for the attribute-set definition 'bib-1', contained in Appendix C:

{ iso member-body US ANSI-standard-Z39.50
attribute-set (3) bib-1 (1) }

A.4 Diagnostic Set

This standard assigns the following object identifier for the diagnostic set definition contained in appendix D.

{ iso member-body US ANSI-standard-Z39.50
diagnostic-set (4) bib-1 (1) }

A.5 Abstract Record Syntax

Object identifier for Record syntaxes are contained in appendix E.

A.6 Transfer Syntax for Non-bibliographic Records

No transfer syntaxes for non-bibliographic records are designated within this standard.

A.7 Resource Report Format

This standard assigns the following object identifier for the resource report format defined in appendix F.

{ iso member-body US ANSI-standard-Z39.50
resource-report-format (7) bib-1 (1) }

APPENDIX B: Definition of Application Context basic-Z39.50-ac

This appendix is part of the standard.

This appendix defines the application context basic-Z39.50-ac. The object identifier for application context basic-Z39.50-ac is registered in Appendix A.

Definition of application context basic-Z39.50-ac
ANSI-standard-Z39.50 application context basic-Z39.50-ac, supports an application-entity that contains only the following two application service elements (ASEs):

- a) the association control service element (ACSE, ISO 8650), and
- b) the Z39.50 service element.

Z39.50 and ACSE are used according to the procedures in section 4.2.1.

The P-Data service is used to transmit Z39.50 APDUs.

In the event of protocol errors, the system detecting the error shall abort the association.

APPENDIX C: Attribute Set bib-1

This appendix is part of the standard.

This Appendix defines the attribute-set bib-1. Attribute-set bib-1 is primarily intended for use with bibliographic databases. The object identifier for attribute set bib-1 is defined and registered in Appendix A. When the value of AttributeSetId (within the

RPNDefinition within the Search APDU) equals the object identifier for this attribute-set, then:

- AttributeType (within AttributeElement within AttributeList) takes values from table C-1 below, and
- AttributeValue takes values from the corresponding attribute table (C2 through C7).

Table C-1: Attribute Types

Attribute Type	Value
Use	1
Relation	2
Position	3
Structure	4
Truncation	5
Completeness	6

Table C-2: Use Attributes

Use	Value	Use	Value	Use	Value
Personal name	1	Date of publication	31	Microform generation	61
Corporate name	2	Date of acquisition	32	Abstract	62
Conference name	3	Title key	33	Note	63
Title	4	Title collective	34	Author-title	1000
Title series	5	Title parallel	35	Record type	1001
Title uniform	6	Title cover	36	Name	1002
ISBN	7	Title added title page	37	Author	1003
ISSN	8	Title caption	38	Author-name personal	1004
LC card number	9	Title running	39	Author-name corporation	1005
BNB card number	10	Title spine	40	Author-name conference	1006
BGF number	11	Title other variant	41	Identifier - standard	1007
Local number	12	Title former	42	Subject - LC children's	1008
Dewey classification	13	Title abbreviated	43	Subject name - personal	1009
UDC classification	14	Title expanded	44	Body of text	1010
Bliss classification	15	Subject precis	45	Date/time added to database	1011
LC call number	16	Subject rswk	46	Date/time last modified	1012
NLM call number	17	Subject subdivision	47	Authority and format identifier	1013
NAL call number	18	Number natl bibliography	48	Concept-text	1014
MOS call number	19	Number legal deposit	49	Concept-reference	1015
Local classification	20	Number govt publication	50		
Subject heading	21	Number publisher for music	51		
Subject Rameau	22	Number db	52		
BDI index subject	23	Number local call	53		
INSPEC subject	24	Code - language	54		
MESH subject	25	Code - geographic area	55		
PA subject	26	Code - institution	56		
LC subject heading	27	Name and title	57		
RVM subject heading	28	Name geographic	58		
Local subject index	29	Place publication	59		
Date	30	CODEN	60		

Table C-3: Relation Attributes

<u>Relation</u>	<u>Value</u>
less than	1
less than or equal	2
equal	3
greater or equal	4
greater than	5
not equal	6

Note: "term" is the right operand of the relation. For example, if 'Use' is 'date', 'relation' is 'less than', and 'term' is '1800', then the relationship is "date less than 1800"

Table C-4: Position Attributes

<u>Position</u>	<u>Value</u>
first in field	1
first in subfield	2
any position in field	3

Table C-5: Structure Attributes

<u>Structure</u>	<u>Value</u>
Phrase	1
word	2
key	3
year	4
date (normalized)	5
word list	6
date (un-normalized)	100
name (normalized)	101
name (un-normalized)	102

Table C-6: Truncation Attributes

<u>Truncation</u>	<u>Value</u>
Right Truncation	1
Left truncation	2
Left and right	3
Do not truncate	100
Process # in the search term	101

Table C-7: Completeness Attributes

<u>Completeness</u>	<u>Value</u>
incomplete subfield	1
complete subfield	2
complete field	3

APPENDIX D: Diagnostic Set bib-1

This appendix is part of the standard.

This Appendix defines the diagnostic set bib-1. The object identifier for diagnostic set bib-1 is defined and registered in Appendix A.

When the value of DiagnosticSetId (within DiagRec within the auxiliary definitions for Search Response and Present Response APDUs) equals the object identifier for this diagnostic set, then "condition" (within DiagRec) takes values from the "Condition" column in the table below.

Table D-1: Diagnostic Conditions

<u>Condition</u>	<u>Meaning</u>	<u>Addinfo</u>	<u>Type</u>
1	Permanent system error		1
2	Temporary system error		1
3	Unsupported search		2
4	Terms only exclusion (stop) words		2
5	Too many argument words		2
6	Too many boolean operators		2
7	Too many truncated words		2
8	Too many incomplete subfields		2
9	Truncated words too short		2
10	Invalid format for record number (search term)		2
11	Too many characters in search statement		2
12	Too many records retrieved		2
13	Present request out-of-range		3
14	System error in presenting records		4
15	Record not authorized to be sent intersystem		4
16	Record exceeds Preferred-message-size		4
17	Record exceeds Maximum-record-size		4
18	Result set not supported as a search term		2
19	Only <u>single</u> result set as search term supported		2
20	Only <u>ANDing</u> of a <u>single</u> result set as search term supported		2
21	Result set exists and replace indicator off		2
22	Result set naming not supported		2
23	Combination of specified databases not supported		2
24	Element set names not supported		1
25	Specified element set name not valid for specified database		1
26	Only a single element set name supported		1
27	Result set no longer exists - unilaterally deleted by target		1
28	Result set is in use		1
29	One of the specified databases is locked		1
30	Specified result set does not exist		1
31	Resources exhausted - no results available		2
32	Resources exhausted - unpredictable partial results available		2
33	Resources exhausted - valid subset of results available		2

<u>Condition</u>	<u>Meaning</u>	<u>Addinfo</u>	<u>Type</u>
100	Unspecified error		1
101	Access-control failure		1
102	Security challenge required but could not be issued - request terminated		1
103	Security challenge required but could not be issued - record not included		4
104	Security challenge failed - record not included		4
105	Terminated by negative continue response		1
106	No abstract syntaxes agreed to for this record		4
107	Query type not supported		2
108	Malformed query		2
109	Database unavailable	database name	2
110	Operator unsupported	operator	2
111	Too many databases specified	maximum	2
112	Too many result sets created	maximum	2
113	Unsupported attribute type	type	2
114	Unsupported Use attribute	value	2
115	Unsupported value for Use attribute	term	2
116	Use attribute required but not supported		2
117	Unsupported Relation attribute	value	2
118	Unsupported Structure attribute	value	2
119	Unsupported Position attribute	value	2
120	Unsupported Truncation attribute	value	2
121	Unsupported Attribute Set	oid	2
122	Unsupported Completeness attribute	value	2
123	Unsupported attribute combination		2
124	Unsupported coded value for term	value	2
125	Malformed search term		2
126	Illegal term value for attribute	term	2
127	Unparsable format for un-normalized value	value	2
128	Illegal result set name	name	1
129	Proximity search of sets not supported		2
130	Illegal result set in proximity search	result set name	2
131	Unsupported proximity relation	value	2
132	Unsupported proximity unit code	value	2

Notes for table D-1

1. "Type" is as follows:

- (1) May occur when search-status or present-status = "failure".
- (2) May occur only when search-status = "failure".
- (3) May occur only when present-status = "failure".
- (4) May occur only as a surrogate for a database record.

2. Where an entry occurs in the "addinfo" column for a given condition, it refers to information

associated with that condition; for example, for condition 114 "Unsupported Use attribute", Addinfo specifies "value", referring to the value of the Use attribute which is not supported. These entries in the Addinfo column are recommended to be included within DiagRec within the auxiliary definitions for Search Response and Present Response APDUs.

APPENDIX E: Record Syntaxes

This appendix is part of the standard.

This appendix defines and registers object identifiers for abstract record syntaxes. These identifiers are intended to be used as the values of PreferredRecordSyntax in Search and Present APDUs.

E.1 Abstract Record Syntaxes

E.1.1 Abstract Record Syntaxes

Registered by this Standard

Names are assigned using the ASN.1 notation (ISO 8824) for OBJECT IDENTIFIER values:

ANSI-Z39-50-2 DEFINITIONS ::= BEGIN

Z39-50 OBJECT IDENTIFIER ::= { iso (1) member-body (2) US (840) ANSI-standard-Z39.50 (10003) }

RecordSyntax OBJECT IDENTIFIER ::= { Z39-50 (5) }

Unimarc ::= OBJECT IDENTIFIER {RecordSyntax (1)}
Intermarc ::= OBJECT IDENTIFIER {RecordSyntax (2)}
CCF ::= OBJECT IDENTIFIER {RecordSyntax (3)}
USmarc ::= OBJECT IDENTIFIER {RecordSyntax (10)}
UKmarc ::= OBJECT IDENTIFIER {RecordSyntax (11)}
Normarc ::= OBJECT IDENTIFIER {RecordSyntax (12)}
Librismarc ::= OBJECT IDENTIFIER {RecordSyntax (13)}
Danmarc ::= OBJECT IDENTIFIER {RecordSyntax (14)}
Finmarc ::= OBJECT IDENTIFIER {RecordSyntax (15)}
MAB ::= OBJECT IDENTIFIER {RecordSyntax (16)}
Canmarc ::= OBJECT IDENTIFIER {RecordSyntax (17)}
SBN ::= OBJECT IDENTIFIER {RecordSyntax (18)}
PicamarC ::= OBJECT IDENTIFIER {RecordSyntax (19)}
END

E.1.2 Local Abstract Record Syntaxes

For registration of local abstract record syntaxes, this appendix defines and registers the following object identifier:

LocalSyntax ::= OBJECT IDENTIFIER { RecordSyntax (1000) }

Local abstract record syntaxes will be registered subordinate to this object identifier. Contact the Z39.50 Maintenance Agency for registration procedures.

E.2 Transfer Syntax for Records

For the purposes of Presentation Context negotiation, record syntax is identified by a pair of object identifiers, one for the abstract-syntax and one for the transfer syntax.

For bibliographic records, no object identifiers for transfer syntax are assigned by this standard. However, an object identifier has been assigned (outside of this standard) for the transfer-syntax for bibliographic records defined in ISO 2709:

{iso standard 2709 transfer-syntax (1) character-encoding (1)}

For non-bibliographic records, transfer syntaxes will be registered subordinate to the object identifier {Z39-50 7} (see Appendix A).

APPENDIX F: Resource Report Format bib-1

This appendix is part of the standard.

This Appendix defines the resource report format bib-1. The object identifier for resource report format bib-1 is defined and registered in Appendix A.

ResourceReportFormat-Bib-1

{ iso member-body US ANSI-standard-Z39.50 resource-report-format (7) bib-1 (1) } ::= BEGIN

-- ResourceReport ::= SEQUENCE{

 estimates [1] IMPLICIT SEQUENCE OF Estimate,
 message [2] IMPLICIT VisibleString}

-- Estimate ::= SEQUENCE{

 type [1] EstimateType,
 value [2] IMPLICIT INTEGER, -- the actual estimate
 currency-code [3] IMPLICIT INTEGER OPTIONAL}
 -- code for representation of currencies defined in ISO 4217-1990.
 -- Applicable only to monetary estimates.

-- EstimateType ::= IMPLICIT INTEGER{

 currentSearchRecords (1), -- estimated number of records in current (incomplete)
 -- result set for search
 finalSearchRecords (2), -- estimated number of records that will be in the result
 -- set for search if the search completes
 currentPresentRecords (3), -- estimated number of records in current (incomplete) set of
 -- records to be returned on Present
 finalPresentRecords (4), -- estimated number of records that will be in the set of records
 -- to be returned by Present if Present completes
 currentOpTimeProcessing (5), -- processing time (in .001 CPU seconds) used by
 -- this operation so far
 finalOpTimeProcessing (6), -- estimated total processing time (in .001 CPU seconds) that will
 -- be used by this operation if it completes
 currentAssocTime (7), -- estimated processing time used by this association
 -- (in .001 CPU seconds)
 currentOperationCost (8), -- estimated cost for this operation so far
 finalOperationCost (9), -- estimated cost for this operation if it completes
 currentAssocCost (10), -- estimated cost for this association so far
 finalOpTimeElapsed (11)}, -- estimated elapsed time for this operation if it completes
 -- (in .001 second units)

END

APPENDIX G: Proximity Query

This appendix is part of the standard.

This Appendix defines the Proximity query, Type-101.

G.1 Structure of the Proximity Query The Proximity query is an extension of the RPN-Query. It is identical in structure with the exception that a proximity operator is represented in addition to the existing RPN-Query operators AND, OR, AND-NOT. Thus the Proximity query is defined as follows:

Proximity-query ::= argument |
 argument + argument + operator
argument ::= operand | Proximity-query
operand ::= attribute-set + term |
 Result-set-id
operator ::= AND | OR | AND-NOT | PROX

Where the notation is as described in section 3.2.2.1.1.1.

G.2 Representation and Evaluation At the origin, the Type-101 query is represented by a tree. Each leaf node contains a simple operand and each non-leaf node contains a complex operand. A simple operand is either a Result-set-id or an Attribute-set + Term. A complex operand is a subtree whose root is an operator and which contains two subtrees: a left-hand operand and a right-hand operand, LO and RO. A complex operand is thus one of the following:

- LO AND RO, representing the intersection of LO and RO;
- LO OR RO, representing the union of LO and RO;
- LO AND-NOT RO, representing the set of elements in LO that are not in RO; or
- LO PROX RO, representing the set of elements resulting from the application of the PROX operator to the operands LO and RO, as described in G.3.

At the target, evaluation of the tree is illustrated by the use of a stack. The tree is processed according to a left post-order traversal. Each leaf-node is one of the following:

- a) Attribute-set + Term
- b) Result-set-id
- c) Operator

Whenever (a) or (b) is encountered, it is put on the stack.

Whenever (c) is encountered:

- Let A and B refer to the last two items put on the stack in the order that they were put on

the stack. A and B are pulled off and the operator is applied as follows:

- o If the operator is AND, OR, or AND-NOT: if A is of the form "Attribute Set + Term", then A is evaluated against the collection of databases specified in the Search request, and the result is A'; otherwise A' = A. Likewise if B is of the form "Attribute Set + Term", then B is evaluated against the collection of databases specified in the Search request, and the result is B'; otherwise B' = B.

Note: in all cases, A' and B' are sets. Each is either a result set or an intermediate set resulting from a previous operation.

- if operator is AND, the result is the intersection of A' and B';
- if operator is OR, the result is the union of A' and B';
- if operator is AND-NOT, the result is the set of elements in A' which are not in B'.

- o If the operator is PROX:

- if both A and B are of the form "Attribute Set + Term" the result is the subset of elements in the collection of databases specified in the Search request, for which A PROX B is true (see G.3).
- If either A or B is not of the form "Attribute Set + Term" (i.e. if A or B is a set) then this standard does not specify the result (see note below).

Note: The use of the proximity operator when one or both operands is a set is for further study. A query that attempts to use the proximity operator in this manner is not considered to be in error. A target that supports the type-101 query may support or not support this use of the proximity operator. If so, the target must also specify the meaning of the result. If not, upon receipt of such a query, the target may return a diagnostic such as "Proximity search of sets not supported" (see appendix D).

- The resulting set is then put on the stack.

When evaluation of the query is complete (i.e. all query-terms have been processed) there will be one item remaining on the stack (otherwise the query is in error) which is the result of the query.

G.3 Proximity Operator The proximity operator, PROX, includes a Distance, Relation-type, Proximity-unit-code, and two boolean flags: Exclusion and Ordered.

Example: suppose A and B respectively specify "personal name = 'McGraw, J.' " and "personal name = 'Stengel, C.' ", and:

- Distance is 0,
- Relation-type is "equal", and
- Proximity-unit is "paragraph".

Then the result is the set of record in which both of the personal names occur within the same paragraph.

Using the same example, if the Exclusion flag is set to "true", the result is the set of record in which the two personal names never both occur within the same paragraph.

If the Ordered flag is set to "true" (and Exclusion flag to "false") then the result is the set of record in which the personal name 'McGraw, J.' occurs within the same paragraph as, but before, the personal name 'Stengel, C.'.

If distance is instead 1 ("ordered" and "exclusion" flag "false") the result is the set of records in which the two personal names occur in adjacent paragraphs. If, in addition, Relation-type is "less-than-or-equal" the result is the set of records in which the two names occur within the same or adjacent paragraphs.

G.4 ASN.1 Description of Proximity Query The proximity query is an extension of the RPN query and has the same structure with the addition of a proximity operator. Thus the proximity query is described by replacing (within the description of RPN-Query in section 4.1.1):

```
Operator ::= [46] CHOICE{
    and      [0] IMPLICIT NULL,
    or       [1] IMPLICIT NULL,
    and-not  [2] IMPLICIT NULL}
```

With the following:

```
Operator ::= [46] CHOICE{
    and      [0] IMPLICIT NULL,
    or       [1] IMPLICIT NULL,
    and-not  [2] IMPLICIT NULL,
    prox     [3] IMPLICIT ProximityOperator}
```

```
ProximityOperator ::= SEQUENCE{
    exclusion    [1] IMPLICIT BOOLEAN
                  OPTIONAL,
    distance     [2] IMPLICIT INTEGER,
    ordered      [3] IMPLICIT BOOLEAN,
    relationType [4] IMPLICIT INTEGER{
        lessThan      (1),
        lessThanOrEqual (2),
        equal          (3),
        greaterThanOrEqual (4),
        greaterThan    (5),
        notEqual       (6)},
    proximityUnitCode [5] CHOICE{
        known      [1] IMPLICIT
                    KnownProximityUnit
        private    [2] IMPLICIT
                    INTEGER}
```

```
KnownProximityUnit ::= INTEGER{
    character (1),
    word      (2),
    sentence  (3),
    paragraph (4),
    section   (5),
    chapter   (6),
    document  (7),
    field     (8),
    subfield  (9),
    fieldtype (10)}
    - Other codes may be added.
```